



ESCOLA NAVAL

talento e biefaire



Chadli Amir

***Comando e controlo remoto de uma
embarcação***

Controlo diferencial dos motores elétricos

**Dissertação para obtenção do grau de Mestre em Ciências Militares
Navais, na especialidade de Engenheiros Navais Ramo de Armas e
Eletrónica**



**Alfeite
2015**



Chadli Amir

Comando e controlo remoto de uma embarcação

Controlo diferencial dos motores elétricos

Dissertação para obtenção do grau de Mestre em Ciências

Militares Navais, na especialidade de

[Engenheiros Navais Ramo de Armas e eletrónica]

Orientação de: CFR EN-AEL Ribeiro Correia

Co-orientação de: CFR EN-MEC (ACN) Pires da Silva

O Aluno Mestrando

O Orientador

[Chadli Amir]

[CFR EN-AEL Ribeiro Correia]

Alfeite

2015

Dedicatória

À minha família e amigos.

À todos que me ajudaram para realizar este trabalho.

Agradecimentos

Quero desde já demonstrar o meu agradecimento:

Ao meu orientador CFR EN-AEL Ribeiro Correia por me ter acompanhado e ajudado a realizar esta dissertação.

A todo o Departamento de Formação de Engenharia Naval de Armas e Eletrónica da Escola Naval, pelo apoio e formação prestada, em especial ao Sargento-mor Mela por todo empenho e tempo gasto neste projeto.

A todos aqueles que ao longo dos anos sempre me apoiaram e tornaram esta dissertação possível.

No âmbito do projeto ICARUS pretende-se desenvolver um veículo autónomo de salvamento marítimo. O objetivo é que este constitua o primeiro socorro de náufragos. Enquadrado neste projeto, proponho-me ao comando e ao controlo diferencial dos dois motores elétricos com que o veículo irá ser equipado sendo que estes irão levar a própria plataforma para o lugar do náufrago. O veículo pode chegar ao lugar do náufrago usando dois modos, o primeiro é com comando à distância com a assistência de um operador, e o segundo será feito em modo autónomo sem ter assistência de nenhum operador. O modo autónomo será feito com auxílio a vários *WayPoints*, ou varias pernadas, durante o caminho até chegar ao local pretendido, com a tensão e autonomia das baterias dos motores e da plataforma em si. Assim, fazer passar o veículo por vários pontos de forma autónoma requer um algoritmo muito eficiente e eficaz, e capaz de prevenir e corrigir todos os erros devido a fatores externos que possam surgir no percurso e que possam dificultar ou impossibilitar a chegada ao ponto desejado, por exemplo os efeitos do vento e da corrente, que são os mais comuns. Este algoritmo será capaz de conjugar os regimes dos dois motores de forma diferencial para conseguir orientar o veículo fazendo as guinadas necessárias e corrigindo também o afastamento ao rumo base devido aos fatores externos. Este algoritmo irá ser inserido no microcontrolador do todo o sistema (*Arduíno*) que por sua vez irá interagir com o recetor do comando remoto de um lado e com o controlador do motor de outro lado, sendo que irá ter, também, um GPS e uma giro bússola como sensores, que serão necessários na parte da orientação e da correção do veículo.

Palavras-chave: Projeto Icarus, Salvamento marítimo, Motores elétricos, Comando e controlo, Diferencial.

ABSTRACT

Key-Words:

Índice

Dedicatória.....	i
Agradecimentos	iii
Resumo	v
ABSTRACT.....	vii
LISTA DE FIGURAS	xii
LISTA DE TABELAS	xv
LISTA DE EQUAÇÕES	xvii
Capítulo 1 – Introdução	18
1.1. Enquadramento do tema.....	18
1.2. Objetivo da dissertação	18
1.3. Estrutura da dissertação	19
Capítulo 2- Estado da arte e conceitos gerais do sistema	21
2.1- Introdução ao Controlo	21
2.2- Ações básicas de controlo	22
Controladores de duas posições (ON-OFF).....	22
Controladores proporcionais (P)	23
Controladores do tipo integral (I).....	23
Controladores do tipo proporcional + integral (P+I)	24
Controladores do tipo proporcional + derivativo (P+D)	25
Controladores do tipo Proporcional + Integral + Derivativo (P+I+D).....	26
2.3- Tipos de controladores	27
Controladores eletrónicos	27
Controladores pneumáticos e hidráulicos.....	28
Ações básicas de controlo VS Tipos de controladores	29
2.4.Descrição de um sistema de controlo diferencial	30
2.5– Arquitetura básica de um sistema de controlo diferencial.....	31
2.6- Matemática de navegação	32
Distância entre dois pontos.....	32
Azimute entre dois pontos	33
Cross-track distance.....	34
Along-track distance	34
2.7 - Estudos relacionados	36
Desenvolvimento de uma capsula telecomandada para as operações de busca e salvamento no mar [15]	36
Estudo simulado de um sistema diferencial eletrónico para veículos elétricos de tração às duas rodas [16].....	38

Desenvolvimento de algoritmos de navegação com <i>WayPoints</i> para os helicópteros Quadrotor [17].....	40
Desenvolvimento de controladores de navegação de baixo custo para os veículos autônomos terrestres [18].....	42
2.8- Síntese dos estudos referidos.....	44
2.9- Especificação dos requisitos do sistema.....	45
Requisitos de <i>hardware</i>	46
Requisitos de <i>software</i>	47
Modos de funcionamento	48
Capítulo 3 – Proposta da Arquitetura do sistema.....	49
3.1. <i>Hardware</i> usado no sistema de comando e controlo.....	49
3.1.1. Motor elétrico <i>Torqueado</i>	49
3.1.2. Placa de desenvolvimento Arduíno	51
3.1.3.Cálculo da curva de velocidade dos motores	52
3.1.4. Placa de interface	54
3.1.5. Sensores.....	56
3.1.6. Comando remoto.....	59
3.2. Algoritmo usado no sistema de comando e controlo.....	61
3.2.1. Inicialização do sistema	62
3.2.2. Processamento de dados de navegação	63
3.2.3. Algoritmo de navegação	69
Capítulo 4 - Testes e Resultados	71
4.2. Simulação de condições de navegação.....	71
Condições e objetivos dos testes.....	71
Descrição dos testes efetuados	73
Análise dos resultados obtidos	74
Conclusões e considerações	77
4.3. Testes de navegação da embarcação	77
Preparações para os testes.....	78
Condições e objetivos dos testes.....	80
Análise dos resultados obtidos	81
Conclusões e considerações	81
Capítulo 5 – Conclusões	82
5.1.Introdução	82
3.2.Revisão dos objetivos traçados.....	82
3.3. Desenvolvimentos futuros.....	84
Referências Bibliográficas	86

Anexo A: Relatório de Atividade	87
Anexo B- Arduíno	88
Anexo C- Código final.....	93

LISTA DE FIGURAS

Fig. 1: Diagrama de blocos de um sistema de controlo automático.....	21
Fig. 2:a) Diagrama de blocos de um controlador ON-OFF (liga desliga); b)Diagrama de blocos de um controlador “ON-OFF” com histerese diferencial.....	23
Fig. 3:Diagrama de blocos de um controlador proporcional	23
Fig. 4:Diagrama de blocos de um controlador integral.....	24
Fig. 5: Variação da saída $u(t)$ em relação a entrada $e(t)$	24
Fig. 6: Diagrama de blocos de um controlador PI	25
Fig. 7: diagrama de blocos de um controlador PD	25
Fig. 8: Diagrama de blocos de um controlador PID	26
Fig. 9: Esquema simplificado de um controlador eletrónico analógico PID	28
Fig. 10: Campos de aplicação dos sistemas hidráulicos e pneumáticos [7]	29
Fig. 11: controlador pneumático P+I.....	29
Fig. 12: Engrenagem de um diferencial mecânico	31
Fig. 13: sistema de controlo diferencial de dois motores.....	32
Fig. 14: Representação de cross and along-track distance	34
Fig. 15: arquitetura do sistema implementado na capsula	38
Fig. 16: Sistema de controlo e de propulsão do EV	39
Fig. 17: Estrutura do sistema de controlo diferencial eletrónico.....	40
A figura 18 apresenta o princípio funcional do controlo de sistema proposto para o diferencial elétrico. Um método de controlo de feedback de <i>loop</i> duplo é adotado para o controlo do motor. O controlo de <i>loop</i> interior controla cada corrente de cada motor e os controlos de <i>loop</i> exteriores controlam a tração do motor para ajustar a velocidade de cada motor. Um microprocessador e vários cartões de interface eletrónica foram usados para implementar o diferencial eletrónico. Um sensor de voltagem e de corrente é usado para cada motor, providenciados os sinais necessários para a estima da velocidade de rotor.	40
Fig. 19: O Quadrotor a seguir a sua trajetória	41
Fig. 20: Trajetória composta por vários WayPoints	41
Fig. 21: GPS ligado ao microcontrolador.....	42
Fig. 22: Subsistema eletrónico do UGV.....	43
Fig. 23: Diagrama do Hardware que será utilizado.....	49
Fig. 24: b)-controlador de velocidade, c)-cabo de ligação entre o controlador de velocidade e o motor, d)-ligação do motor s baterias, e)-modulo GPS, f)-motor, g)-baterias (Jesus,2013).	50
Fig. 25: Esquema geral da comunicação da alavanca com o motor	50
Fig. 26: Sistema implementado substituto do sensor magnético.....	52
Fig. 27: A sequência binaria DO com várias posições da alavanca	52
Fig. 28: Curva de velocidade do motor	53
Fig. 29: Esquemático do circuito lógico.....	55
Fig. 30: Circuito impresso final.....	55
Fig. 31: Bússola modelo COMPS03	57
Fig. 32: Montagem para ler a bússola.....	57
Fig. 33: GPS modelo <i>GlobalSat EM 406</i>	58
Fig. 34: Montagem para ler o GPS	58
Fig. 35: Demonstração das frases lidas pelo GPS e a informação que contém	59
Foi utilizada a função <i>PulseIn</i> para ler os sinais <i>PWM</i> dos canais enviados do transmissor para o recetor. Esta função Lê um pulso (tanto HIGH como LOW) em um pino. Por exemplo, se valor for HIGH, <i>pulseIn()</i> espera que o pino vá para HIGH, inicia a cronometragem, e então espera que o pino vá para LOW e para a cronometragem. Retorna a duração do pulso em microssegundos. Para variar a velocidade dos motores é necessário alterar o “valor”, alterando	

assim a largura de pulso e consequentemente o Duty Cycle. A Figura 36 apresenta vários Duty Cycle's que o recetor possa receber do transmissor.	59
Fig. 37: montagem para ler os sinais dos canais do recetor	59
Fig. 38: Calcular a velocidade a partir dos sinais PWM.....	60
Fig. 39: Pulse Width Modulation e Duty Cycle.....	60
Fig. 40:Princípio de funcionamento do telecomando.....	60
Fig. 41: telecomando para fazer o controlo da embarcação	61
Fig. 42: Fluxograma geral do sistema de controlo autónomo	62
Fig. 43: Fluxograma do bloco “inicializações”	63
Fig. 44: Bloco de processamento de dados e navegação 1.....	64
Fig. 45: Bloco de processamento de dados e navegação 2.....	65
Fig. 46: Bloco de processamento de dados e navegação 3.....	66
Fig. 47: Trajetórias a evitar	68
Fig. 48: como o algoritmo recalcula um novo rumo para a embarcação	68
Fig. 49: Fluxograma da comparação	69
Fig. 50: Protótipo utilizado.....	72
Fig. 51: Durante a realização do percurso	73
Fig. 52: WP's e o percurso.....	73
Fig. 53: O Segundo percurso efetuado na parada escola naval	Erro! Marcador não definido.
Fig. 54: Embarcação a guinar para BB.....	75
Fig. 55: Embarcação a guinar para EB.....	75
Fig. 56: Embarcação a seguir para frente.....	76
Fig. 57: Algoritmo a incrementar o contador dos WP's.....	76
Fig. 58: Algoritmo a mandar parar os motores.....	77
Fig. 59: Embarcação no seu estado inicial	78
Fig. 60: Embarcação já com motores montados.....	78
Fig. 61: Protótipo de controlo Arduino + Sensores.....	78
Fig. 62: Posição do protótipo na embarcação e na caixa com o resto dos equipamentos.....	78
Fig. 63: Alimentação do Arduino.....	79
Fig. 64: Alimentação dos motores	79
Fig. 65: Alimentação das placas de interface.....	79
Fig. 66: A caixa onde se interligam todos os equipamentos.....	79
Fig. 67: Embarcação na água	79
Fig. 68: Trajetória escolhida para fazer os testes.....	81
Fig. 69: Foto da trajetória feita na primeira tentativa.	Erro! Marcador não definido.
Fig. 70: trajetória realizada na segunda tentativa.	Erro! Marcador não definido.
Fig. 71: Exemplo de embarcações usadas no mesmo âmbito de trabalho (Projeto Icarus)	85
Fig. 72: Arduino Duemilanove.....	89
Fig. 73: Arduino Mega 2560.....	89
Fig. 74: Microcontrolador vs Microprocessador.....	89

LISTA DE TABELAS

Tabela 1: Matriz de comparação entre os vários tipos de controlo	30
Tabela 2: Tipo de Memória e utilização dada	90
Tabela 3: Quantidade de memória nos microcontroladores ATmega168 e Atmega2560	90
Tabela 4: PinOut dos Modelos Duemilinue e Mega2560.....	91

LISTA DE EQUAÇÕES

Equação 1: Equação que define o controlador do tipo(ON-OFF)	22
Equação 2: Relação que define o controlador do tipo (P)	23
Equação 3: Relação que define o controlador do tipo (I)	23
Equação 4: Relação que define o controlador do tipo (P+I)	24
Equação 5: Relação que define o controlador do tipo (P+D).....	25
Equação 6: Relação que define o controlador do tipo (P+I+D).....	26
Equação 7: Aproximação equiratangular	33
Equação 8: Lei de cossenos esférica	33
Equação 9: Fórmula de Haversine [5]	33
Equação 10: Equação de azimuth entre dois pontos [5]	33
Equação 11: Equação cross track distance [5]	34
Equação 12: Equação along-track distance [5]	35

1.1. Enquadramento do tema

A busca e salvamento no mar é uma complicada e perigosa tarefa numa situação de desastre, devido a grande área de busca e número de pessoas em risco. O aproveitamento do avanço tecnológico alcançado da robótica móvel e veículos não tripulados nos laboratórios e centros de pesquisa melhora o desempenho das equipas no campo e reduz o seu tempo de reação.

O objetivo do projeto ICARUS é aproveitar os benefícios deste avanço tecnológico e pô-lo em prática para salvar a vida humana, através de desenvolvimento de veículos terrestres, *Drones* e embarcações autónomas que podem operar individualmente ou cooperar juntos através de sistemas cujo seu desenvolvimento é um dos objetivos do projeto. Estes robots autónomos são a primeira ação na tarefa das equipas de salvamento sendo que a sua função é armazenar e fornecer os dados necessários sobre a área para o centro de operações, bem como procurar a vítima e dar a ajuda necessária às pessoas em risco no local.

As embarcações autónomas fazem parte do objetivo do projeto ICARUS devido às características ambientais e geográficas especiais do mar, tais como a grande área de busca em caso de navios cumulativamente com a dispersão náufragos, limitações que o mau tempo provoca para as operações de busca e salvamento, e o estreito intervalo de tempo disponível para qualquer ação. Um dos requisitos do projeto é o desenvolvimento de uma cápsula autónoma de salvamento que vai fazer parte da embarcação, capaz de navegar no mar de modo a cobrir uma determinada área e juntar os dados necessários sobre o local, e procurar vítimas de forma eficaz. Neste sentido vem a importância e a necessidade de ter um sistema de controlo e de comando que efetua diretamente nos motores que equiparão a embarcação para que ela consiga fazer o que é pretendido de forma eficaz.

O objetivo da dissertação é equipar a embarcação não só com os motores mas com sistema completo de comando e controlo que consiga trabalhar em dois modos (autónomo e comando a distancia), tendo como grande preocupação os vários fatores externos que afetam a embarcação e a própria autonomia das baterias dos motores.

1.2. Objetivo da dissertação

Pretende-se fazer um sistema de controlo de uma embarcação com uma arquitetura e características específicas para cumprir com os objetivos traçados no projeto ICARUS. Esta embarcação vai ser estudada pormenorizadamente com a sua arquitetura e as respetivas características na tese do CAD EN-MEC Brahimi Younes. Em avanço à existência de especificações técnicas da embarcação no que se refere ao seu sistema propulsor, utilizar-se-ão como meios de experimentação, os motores propulsores elétricos e uma embarcação existentes no CINAV.

O sistema que vai ser implementado vai ter em conta que a nossa embarcação não vai ter leme a bordo para facilitar a manobra, ou seja vai-se jogar com os motores elétricos (BB e EB) por forma a orientar eficazmente a embarcação, sendo que daí surge o problema do controlo diferencial destes motores que consiste em dar vários regimes diferentes para cada um dos dois motores de modo a fazer as respetivas manobras e guinadas, de forma antecipada e precisa.

É de salientar que se vai ter em consideração, também, o Vento e a Corrente que vão influenciar bastante o rumo desejado, portanto vai haver uma correção contínua do tal rumo, ou seja vai haver um tipo de comparação contínua entre aquele que é suposto fazer e aquele que a ser feito e efetuar as tais correções necessárias para estar sempre dentro do rumo principal.

1.3. Estrutura da dissertação

Esta tese descreve todas as decisões e abordagens tomadas de forma a conclui-la, sendo que alguns pormenores foram omissos para tornar a sua leitura mais agradável. A tese está estruturada em cinco capítulos:

Capítulo 2 - Estado da arte e conceitos gerais do sistema – Neste capítulo é feito o estudo teórica do controlo em geral, de seguida é feito um estudo sobre os sistemas e os projetos desenvolvidos no mesmo âmbito, terminando por especificar os requisitos funcionais do sistema que vai ser desenvolvido.

Capítulo 3 – Proposta da arquitetura do sistema – Neste capítulo é feita a descrição do trabalho desenvolvido relacionado tanto com a placa de interface como com a parte do *software* e do Hardware.

Capítulo 4 - Teste e resultados – Aqui é feita a descrição e análise dos testes que foram feitos.

Capítulo 5 – Conclusões – revisão dos objetivos traçados inicialmente e as conclusões tiradas, e por fim trabalhos futuros propostos pelo autor.

Neste capítulo será feita uma introdução sobre os conceitos teóricos do controlo em geral, onde terá especial enfoque no sistema de controlo que se pretende usar neste trabalho de investigação, ou seja controlo diferencial de motores elétricos. A componente principal deste capítulo é uma revisão bibliográfica dos trabalhos realizados na área dos sistemas de controlo e alguns projetos já feitos para realizar o mesmo objetivo. Será também incluída uma secção dedicada à apresentação de algoritmos de controlo que possam ser usados num veículo de forma eficaz e que possam também aproveitar as informações vindas dos sensores para fazer as tais correções. O capítulo termina com a especificação do sistema que se pretende construir, relativamente aos requisitos do *software* e do hardware aproveitando os trabalhos e projetos realizados na área e referidos na secção 2.7.

2.1- Introdução ao Controlo

Um controlador automático tem como função produzir um sinal de controlo que anule o erro (desvio), ou o reduza a um valor muito pequeno. O controlador compara o valor real da saída do processo com o valor desejado (set-point), determina o erro ou desvio, e produz o respetivo sinal de comando para o atuador [1].

Na Fig.1 podemos observar um diagrama de blocos de um controlador industrial, bem como o elemento de medida (sensor).

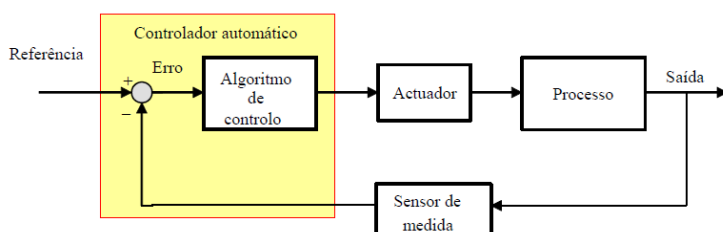


Fig. 1: Diagrama de blocos de um sistema de controlo automático

O controlador é formado por um detetor de erro (ponto de soma) e um algoritmo de controlo + amplificador. O sensor de medida converte o sinal de saída num outro sinal, como seja um deslocamento, pressão ou sinal elétrico que é utilizado para comparar a saída em relação ao sinal de entrada de referência. Este elemento, constitui o ramo de realimentação do sistema em anel fechado. O ponto de ajuste ou o valor de referência (set-point) do controlador que é o valor desejado para a variável de processo, deve ser convertido numa entrada de referência com as mesmas unidades do sinal de realimentação do sensor de medida. O amplificador tem como função amplificar a potência do sinal de saída do

controlador, de modo a poder operar o atuador. O atuador tem como função alterar a entrada do processo (Ex: velocidade, rumo, temperatura) de acordo com o sinal de controlo, de modo a que a saída do processo seja igual ou o mais próximo possível do valor de referência (set-point) [2].

2.2- Ações básicas de controlo

A realização do sinal de comando pelo controlador, pode ser obtida de diversas formas, designadas por "ações de controlo ou de regulação", o que nos permite classificar os controladores da seguinte forma [3] [4] :

- **Controladores de duas posições (ON-OFF)**
- **Controladores proporcionais (P)**
- **Controladores do tipo integral (I)**
- **Controladores do tipo proporcional + integral (P+I)**
- **Controladores do tipo proporcional + derivativo (P+D)**
- **Controladores do tipo proporcional + integral + derivativo (P+I+D)**

Controladores de duas posições (ON-OFF)

Este sistema de controlo apresenta a vantagem de ser simples e barato, o que se traduz na sua grande aplicação, tanto em sistemas industriais como domésticos. Neste sistema, o controlador possui apenas duas posições fixas, que são a de ligado ou desligado. Considerando o sinal de saída do controlador $u(t)$ e o sinal de erro $e(t)$, num controlo deste tipo, o sinal $u(t)$ permanece ou num valor máximo ou num valor mínimo, dependendo do sinal de erro ser positivo ou negativo [4] .Assim:

$$M_1, M_2 = \text{constantes}$$
$$\begin{cases} u(t) = M_1 & \text{para } e(t) > 0 \\ u(t) = M_2 & \text{para } e(t) < 0 \end{cases}$$

Equação 1: Equação que define o controlador do tipo(ON-OFF)

O valor mínimo M_2 é normalmente nulo ou $- M_1$. Os controladores deste tipo são geralmente dispositivos elétricos, onde é bastante utilizada uma válvula operada por um solenoide elétrico (Ex: termostato elétrico de temperatura).

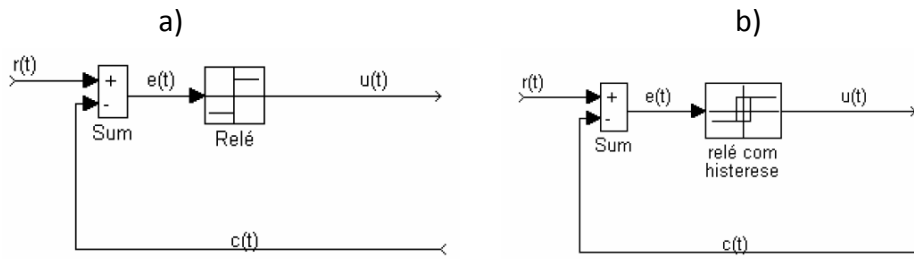


Fig. 2:a) Diagrama de blocos de um controlador ON-OFF (liga desliga); b) Diagrama de blocos de um controlador "ON-OFF" com histerese diferencial

Controladores proporcionais (P)

Neste caso a relação entre a saída $u(t)$ e o sinal de erro $e(t)$, é dada pela expressão:

$$u(t) = k_p \cdot e(t) \Rightarrow U(s) = k_p \cdot E(s)$$

Equação 2: Relação que define o controlador do tipo (P)

Em que K_p é designado por sensibilidade ou ganho proporcional. Um controlador deste tipo, consiste basicamente de um amplificador de ganho ajustável [4] [3], conforme apresentado na Fig. 3.

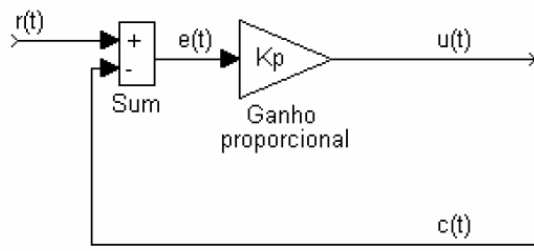


Fig. 3: Diagrama de blocos de um controlador

Este controlador é pouco preciso (precisão baixa), embora a estabilização da saída seja relativamente rápida [3].

Controladores do tipo integral (I)

O valor do sinal de saída do controlador é a soma ou integral do valor do erro, desde um determinado instante [4]. A Fig. 4 Representa um controlador integral.

$$\frac{du(t)}{dt} = K_i \cdot e(t) \quad K_i \rightarrow \text{ganho Integral}$$

$$u(t) = K_i \int_0^t e(t) dt \Rightarrow \frac{U(s)}{E(s)} = \frac{K_i}{s} = \frac{1}{sT_i} \quad T_i \rightarrow \text{tempo Integral}$$

Equação 3: Relação que define o controlador do tipo (I)

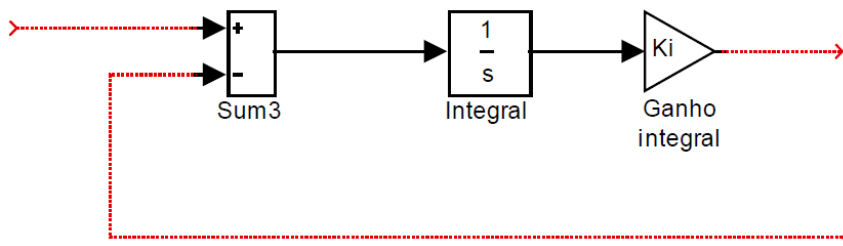


Fig. 4: Diagrama de blocos de um controlador

Como mostra a Fig.5, o valor $e(t)$ ao ser duplicado irá fazer variar duas vezes mais rápido $u(t)$. No caso de $e(t)=0$, o valor de $u(t)$ irá manter-se constante (estacionário), pois $u(t)$ varia no tempo numa taxa que é proporcional ao sinal de erro que atua em $e(t)$ [3] [4].

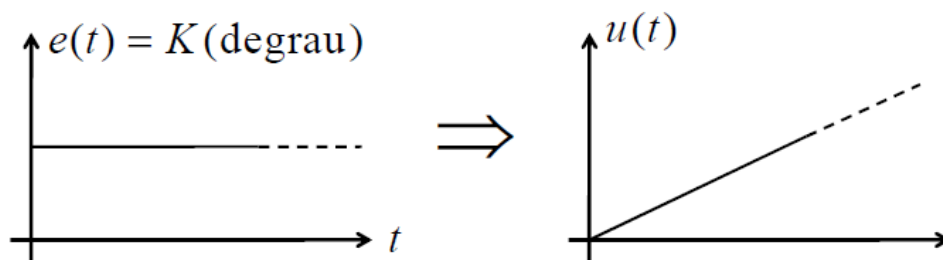


Fig. 5: Variação da saída $u(t)$ em relação a entrada $e(t)$

Controladores do tipo proporcional + integral (P+I)

Esta ação de controlo, é definida pela seguinte equação [3] [4]:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt$$

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) = K_p \left(\frac{s + 1/T_i}{s} \right)$$

Equação 4: Relação que define o controlador do tipo (P+I)

Em que K_p é a sensibilidade ou ganho proporcional e T_i é o tempo integral, corresponde ao tempo em que a parcela relativa a parte proporcional da ação de controlo é duplicada. Tanto K_p como T_i são ganhos ajustáveis. T_i tem como função ajustar a ação de controlo integral, enquanto K_p tem ação sobre a parte proporcional. O tempo integral, aparece em muitos controladores com a escala de minutos por repetição (m.p.r) [5].

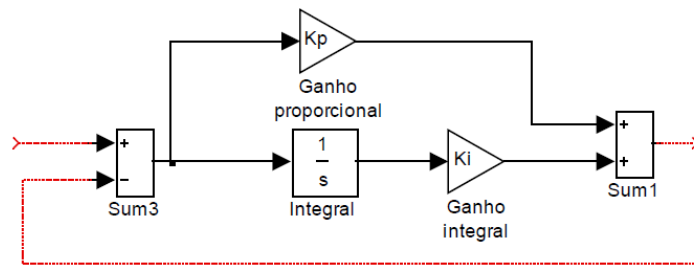


Fig. 6: Diagrama de blocos de um controlador PI

A precisão deste controlador é boa, embora a estabilização da saída seja obtida ao fim de bastante tempo [3].

Controladores do tipo proporcional + derivativo (P+D)

Esta ação de controlo é definida pela seguinte equação [3] [4]:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

$$\frac{U(s)}{E(s)} = K_p (1 + s T_d)$$

Equação 5: Relação que define o controlador do tipo (P+D)

Em que T_d é o tempo derivativo definido em segundos ou minutos. Tanto K_p como T_d são grandezas ajustáveis. A ação de controlo derivativa, é proporcional à taxa de variação do sinal de erro. O tempo derivativo T_d é o intervalo de tempo durante o qual a ação derivativa antecipa o valor da ação de controlo proporcional.

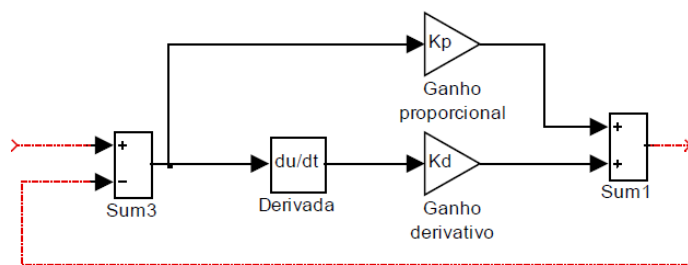


Fig. 7: diagrama de blocos de um controlador PD

O tempo derivativo é definido como o tempo em minutos em que o modo derivativo adianta o efeito do modo proporcional. Quanto maior o tempo derivativo (T_d), mais forte é a ação derivativa.

Controladores do tipo Proporcional + Integral + Derivativo (P+I+D)

Esta ação resulta da combinação de três ações, que são: a proporcional, a derivativa e a integral (PID). A equação respeitante a esta ação mista, é dada por [3] [4]:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$
$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

Equação 6: Relação que define o controlador do tipo (P+I+D)

A equação anterior é também conhecida em controlo industrial por **algoritmo ou equação ideal**.

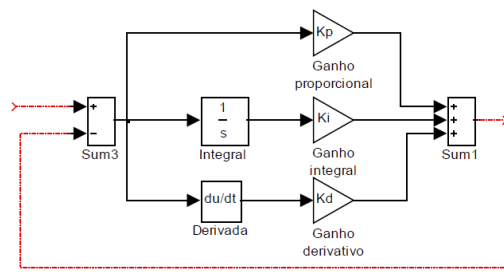


Fig. 8: Diagrama de blocos de um controlador PID

O PID é o algoritmo de controlo mais usado na indústria e tem sido utilizado na maioria dos sistemas de controlo [4]. A popularidade de controladores PID pode ser atribuída em parte ao seu desempenho robusto numa ampla gama de condições de funcionamento e à sua simplicidade funcional. Este controlador permite obter uma estabilização rápida e com um erro em regime estacionário nulo. O ganho proporcional é responsável pela velocidade de resposta do sistema. Quando se aumenta o ganho proporcional, o sistema torna-se mais rápido, mas deve-se ter cuidado para não o tornar instável. Uma vez que P foi definido para obter uma resposta rápida desejada, o termo integral é aumentado a fim de parar as oscilações [5]. O termo integral reduz o erro em regime estacionário, mas aumenta o valor de overshoot. Este valor t é sempre necessário para um sistema rápido de modo que possa responder às mudanças de forma imediata [4]. O termo integral é novamente ajustado para atingir um mínimo erro de *steady state*. Uma vez que o P e I foram definidos para que o sistema de controlo seja rápido com o *steady state* mínimo e constante, o termo derivativo é aumentado até que o loop seja aceitavelmente rápido em relação ao seu ponto de referência. Aumentar o termo da derivada diminui o overshoot, aumentando o ganho, mantendo a

estabilidade e ainda torna o sistema altamente sensível ao ruído. Muitas vezes, existe a necessidade de fazer a compensação de uma característica de um sistema de controlo para melhorar outro, por forma a atender as suas necessidades [3].

Para o nosso trabalho, será importante o sistema PID pois este fornece uma estabilização rápida após as diversas guinadas que terão de ser feitas para o controlo diferencial dos dois motores. Também não permite que o sistema fique a oscilar ou no nosso caso fazer a embarcação andar de forma harmónica, o que não é desejado [5] [3].

2.3- Tipos de controladores

Ao longo deste século, foram surgindo diversos tipos de controladores analógicos cada vez mais sofisticados, com especial destaque para os pneumáticos numa primeira fase e que de certa forma, inauguraram a era moderna do controlo automático industrial [5] [2]. Para além dos controladores pneumáticos, que foram geralmente utilizados no controlo de processos industriais (nível, temperatura, caudal, etc.), os controladores hidráulicos foram também ganhando destaque especialmente no controlo de posição e velocidade (Ex: controlo do passo dos hélices e do ângulo do leme dos navios, velocidade de rotação dos motores, etc.).

Como tal os controladores podem ser classificados de acordo com o tipo de tecnologia utilizada na sua construção [2]. Deste modo, poderemos ter:

- Controladores eletrónicos (analógicos e digitais)
- Controladores pneumáticos
- Controladores hidráulicos

Controladores eletrónicos

Os controladores eletrónicos analógicos atuais, utilizam em larga escala um componente ativo designado por "Amplificador Operacional". As possibilidades de efetuar montagens com estes componentes são bastante elevadas devido à sua versatilidade de aplicação [2]. No campo específico do controlo industrial, são muito utilizadas (entre outras), as seguintes montagens típicas:

- Amplificador diferencial (ponto de soma).
- Amplificador inversor (ação proporcional).
- Amplificador integrador (ação integral).

- Amplificador diferenciador (ação derivativa).
- Amplificador somador (soma de várias ações de controle).

O esquema em baixo representa um controlador analógico PID. Como podemos ver, este controlador eletrônico é composto por vários amplificadores operacionais interligados entre si através várias realimentações feitas com o auxílio das resistências e dos condensadores [6].

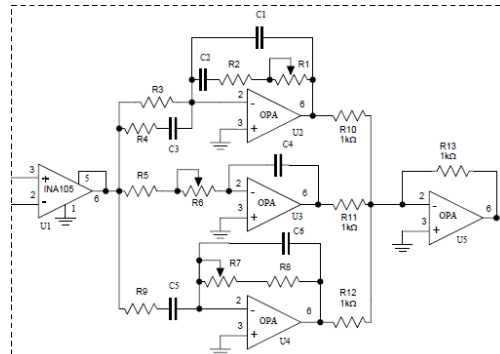


Fig. 9: Esquema simplificado de um controlador eletrônico analógico PID

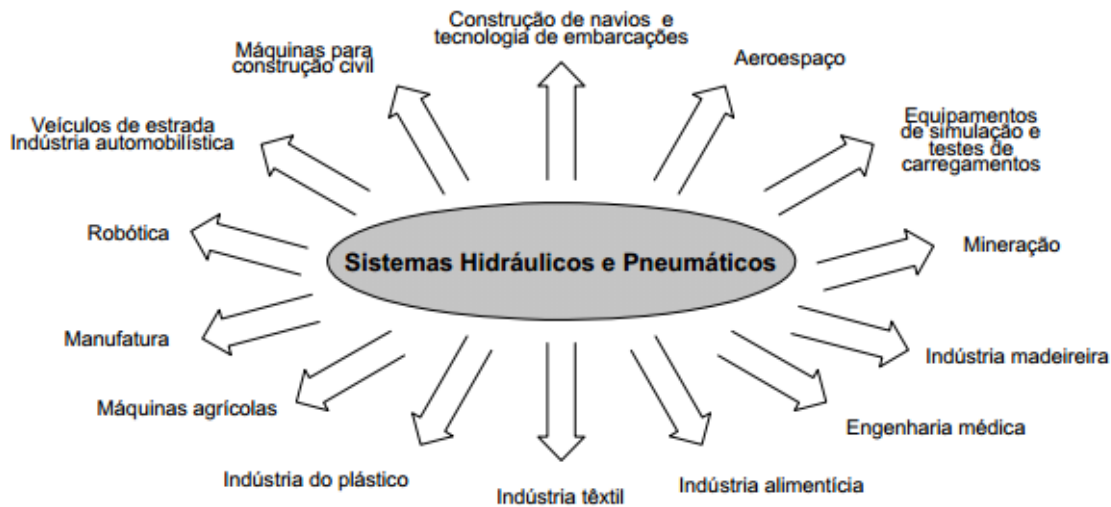
Controladores pneumáticos e hidráulicos

Devido à sua natureza, os sistemas hidráulicos e pneumáticos constituem-se numa forma concreta de aplicação dos princípios da mecânica dos fluidos compressível e incompressível a qual embaça o desenvolvimento de componentes e circuitos [7].

Os controladores pneumáticos baseiam-se no uso de gás e de ar comprimido, o qual é desprovido de propriedades lubrificantes contendo vapor de água, enquanto os hidráulicos baseiam-se no uso de líquidos, neste caso o óleo que funciona como fluido hidráulico e também lubrificante [7] [6].

As potências de saída dos sistemas pneumáticos são consideravelmente menores do que as dos sistemas hidráulicos [6], sendo que a precisão destes é satisfatória em todas as velocidades enquanto a dos pneumáticos diminui nas baixas velocidades [7].

Tal como mostra a figura 10, estes dois sistemas possuem uma vasta aplicação nas mais diversas áreas, sendo que umas das ações mais utilizada por ambos é a “ON-OFF”, mais conhecida por “liga-desliga” [2].



Existem também controladores hidráulicos e pneumáticos do tipo proporcional + integral. A ação integral obtém-se através da adição de mais um fole (fole da ação integral- I) ao dispositivo de forças do controlador proporcional, conforme representado na figura 11. Este fole encontra-se em oposição ao fole de saída (O). A alimentação de ar ao fole I efetua-se através da pressão da saída, mas não é direta, pois o ar tem que atravessar uma válvula (restrição) designada por “a” [7].

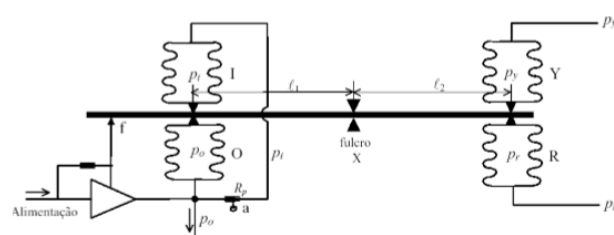


Fig. 11: controlador pneumático P+I

É de salientar que por razões de tempo, não se apresentam os desenvolvimentos relativamente aos controladores pneumáticos PD e PID.

Ações básicas de controlo VS Tipos de controladores

Tabela 1: Matriz de comparação entre os vários tipos de controlo

	ON-OFF	P	I	P+I	P+D	P+I+D
Eletrónico	V.A	V.A	V.A	V.A	V.A	V.A
Pneumático	V.A	V.A	P.A	P.A	R.A	R.A
Hidráulico	V.A	V.A	P.A	P.A	R.A	R.A

Legenda: V.A: Vasta Aplicação; P.A: Pouca Aplicação; R.A: Raramente usado.

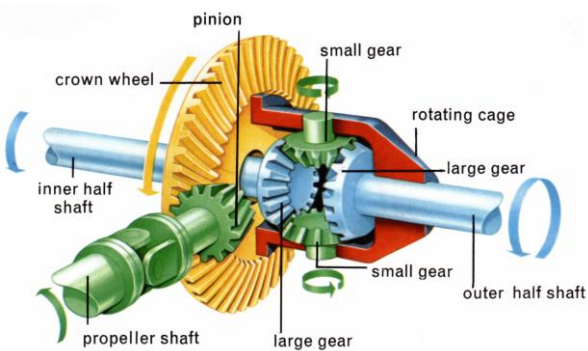
Com o avanço da tecnologia eletrónica, numa primeira fase analógica e mais recentemente digital, esta tecnologia ganhou uma dimensão que hoje em dia a grande maioria das aplicações industriais, recorre quase que exclusivamente a controladores eletrónicos do tipo digital [3] [2] [1].

O nosso sistema precisa de um controlador eletrónico porque é o mais adequado e o mais compatível com o resto dos equipamentos que fazem parte da arquitetura completa do sistema. O tipo de controlador eletrónico que vai ser usado é o PID devido ao seu desempenho robusto numa amplo gama de condições de funcionamento e à sua simplicidade funcional. Este controlador permite obter uma estabilização rápida e com um erro em regime estacionário nulo.

2.4.Descrição de um sistema de controlo diferencial

A palavra diferencial ou sistema diferencial é mais conhecida no âmbito das áreas de mecânica e telecomunicações (transmissão de dados), no entanto só irá ser abordada a primeira, visto que se enquadra mais na problemática do controlo diferencial de motores elétricos.

O grande feito do diferencial mecânico é conseguir compensar a diferença de percurso das rodas numa curva, por meio de engrenagens, sem perder a tração. Quando um veículo faz uma curva, a roda do lado externo da curva percorre uma distância maior que a roda do lado interno. Se essa distância não for compensada, haverá um arrasto da roda que está no lado interno da curva,



provocando desgaste excessivo dos pneus e dificuldade para controlar o veículo. É aí que o diferencial entra em ação [8].

Fig. 12: Engrenagem de um diferencial mecânico

No trabalho de investigação presente neste documento, o sistema de controle em estudo deve ter uma ação diferencial. Porém, esta ação não será realizada por um sistema mecânico. O nome diferencial deve-se ao fato de se tornar necessário fazer uma combinação entre os dois motores elétricos, por forma a conduzir a nossa embarcação corretamente. Num sistema mecânico, quando queremos virar, uma roda terá que rodar mais rápido que a outra para conseguir fazer a tal curva. Assim, no nosso sistema também precisamos de fazer o mesmo mas, em vez de termos duas rodas, vamos ter dois motores elétricos. No entanto, para fazer os motores andarem a velocidades diferentes não vão ser utilizadas as tais engrenagens do diferencial (as quais tornavam o trabalho mais fácil), mas sim um sistema de controle digital, que irá fazer o tal controle diferencial da nossa embarcação. Esse sistema é composto por vários requisitos de Software e de Hardware que vão ser explicados posteriormente mais detalhadamente.

2.5– Arquitetura básica de um sistema de controle diferencial

Por forma a conseguir fazer o controle diferencial dos dois motores, é necessário fazer um deles funcionar com uma potência mais elevada que a do outro, por forma a realizar as guinadas necessárias e compensar os fatores externos (vento e corrente) que possam influenciar o comportamento do veículo. Assim, para o motor efetuar a manobra ou a guinada necessária para orientar a embarcação para o rumo pretendido, independentemente das condições externas que possam influenciar a chegada, é necessário um conjunto de sensores que vão fornecendo relatórios sobre o estado da nossa embarcação em termos da posição, velocidade e atitude, bem como informando acerca do estado externo onde se encontra a embarcação. Todas estas informações serão transmitidas para o controlador que vai interagir

com todas elas, e por conseguinte vai controlar os motores de modo a efetuar as manobras mais adequadas [5] [9] [10].

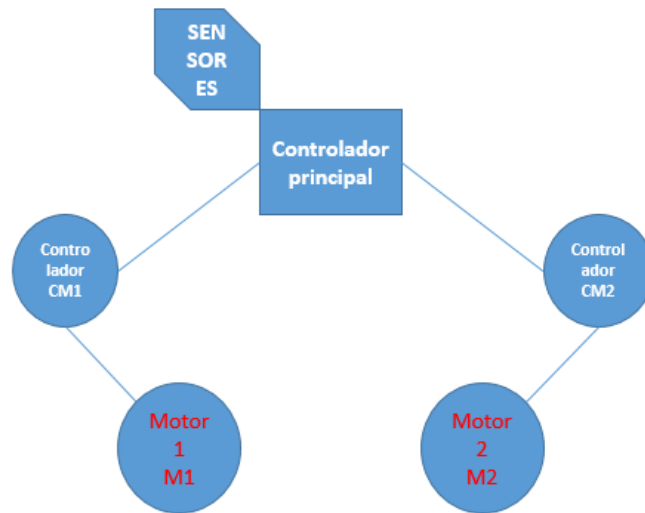


Fig. 13: sistema de controlo diferencial de dois motores

2.6- Matemática de navegação

De forma a executar um plano de navegação é necessário sermos capazes de calcular distâncias, azimutes, velocidades [10]. Contudo e devido à simplicidade que o sistema de controlo visa ter, são apenas abordados os seguintes conceitos:

- Distância entre dois pontos;
- Rumo entre dois pontos;
- *Cross-track distance/error* ou afastamento ao planeamento;
- *Along-track distance* ou distância percorrida.

Salienta-se que para a obtenção dos conceitos acima enunciados não foi feito um estudo profundo sobre a teoria matemática de que resultam, sendo apenas apresentadas as fórmulas de aplicação direta já deduzidas.

Distância entre dois pontos

Existem várias formas de calcular distâncias entre dois pontos no planeta terra. Desde a aproximação à projeção equiretangular deduzida da projeção cilíndrica equidistante [11], a Lei dos Cossenos esférica [12], ou ainda a fórmula de Haversine, simplificação da trigonometria

esférica, muito usada em navegação [13]. Nesta dissertação não serão abordados os métodos mais simples associados a trigonometria plana.

Fórmulas de aplicação direta para cálculo de distância [14]:

$$d = R * \sqrt{((Lf - Ls) * \cos(\frac{\varphi f + \varphi s}{2}))^2 + (\varphi f - \varphi s)^2} ;$$

Equação 7: Aproximação equirangular

$$d = \cos^{-1}(\sin\varphi s \sin\varphi f + \cos Ls \cos Lf \cos(Lf - Ls)) * R$$

Equação 8: Lei de cossenos esférica

$$d = 2 * \tan 2^{-1} \left(\frac{\sqrt{\sin^2 \left(\frac{\varphi f - \varphi s}{2} \right) + \cos Ls \cos Lf \sin^2 \left(\frac{Lf - Ls}{2} \right)}}{1 - \left(\sin^2 \left(\frac{\varphi f - \varphi s}{2} \right) + \cos Ls \cos Lf \sin^2 \left(\frac{Lf - Ls}{2} \right) \right)} \right) * R$$

Equação 9: Fórmula de Haversine [14]

Onde: Lf- longitude do ponto final;

Ls- longitude do ponto inicial

d- distância entre pontos;

R: Raio da terra

φf : Latitude do ponto final.

φs : Latitude do ponto inicial

Para o sistema de navegação vai ser usada a fórmula de *Haversine*. Esta escolha baseou-se em testes efetuados em zonas conhecidas, sendo a confrontação de resultados efetuada com recurso ao *Google Earth*, programando-se a lei de cossenos e a lei de *Haversine* em *Arduíno*. As distâncias obtidas tinham uma resolução de centímetros.

Azimute entre dois pontos

O azimute entre dois pontos é a direção em relação ao Norte entre esses dois pontos e pode ser calculada pela seguinte expressão:

$$d = 2 * \tan 2^{-1} \left(\frac{\sin(Lf - Ls) \cos \varphi f}{\cos \varphi s \sin \varphi f - \sin \varphi s \cos \varphi f \cos(Lf - Ls)} \right)$$

Equação 10: Equação de azimute entre dois pontos [14]

Onde: Lf- Longitude do ponto final

Ls- Longitude do ponto inicial
 Θ - Azimute entre dois pontos;
 ψ_f : Latitude do ponto final
 ψ_s : Latitude do ponto inicial

Cross-track distance

Cross-track distance é a expressão inglesa para afastamento lateral ao planeamento que em relação à linha que une dois pontos.

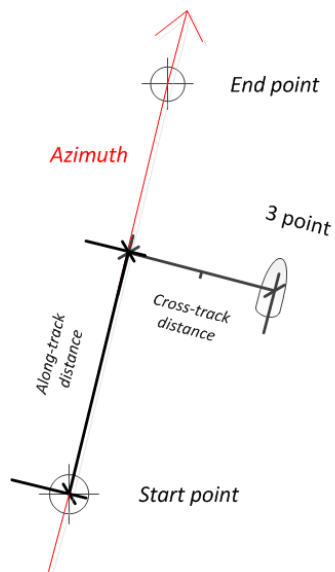


Fig. 14: Representação de cross and along-track distance

Podemos calcular esse afastamento com a seguinte equação:

$$d_{xt} = \sin^{-1} \left(\sin \left(\frac{d_{13}}{R} \right) * \sin(\theta_{13} - \theta_{12}) \right) * R$$

Equação 11: Equação cross track distance [14]

Onde: d_{13} é a distância do ponto inicial ao terceiro ponto

θ_{13} é o rumo entre ponto inicial e terceiro ponto

θ_{12} é o rumo entre ponto inicial e final

R é o raio da Terra

Along-track distance

O *Along-track distance* é a distância que já foi percorrida em relação à linha que une dois pontos, ver figura

10, podendo ser obtida pela seguinte expressão:

$$dat = \cos^{-1} \left(\frac{\cos \left(\frac{d13}{R} \right)}{\cos \left(\frac{dxt}{R} \right)} \right) * R$$

Equação 12: Equação along-track distance [14]

Onde: d13 é a distância do ponto inicial ao terceiro ponto

dxt é a cross-track distance

2.7 - Estudos relacionados

Nesta secção, ir-se-ão abordar alguns dos estudos e projetos que se encontram mais relacionados com o tema desta dissertação, tais como, a implementação de sistemas de controlo em termos de uso de alguns equipamentos específicos e a maneira como foram utilizados e explorados para resolver os problemas encontrados, quais os algoritmos usados, bem como a sua interferência com os requisitos de *hardware*. Pretende-se com a análise destes projetos encontrar soluções que possam ser aplicadas neste trabalho, efetuando as necessárias adaptações por forma a ser obtido o sistema a implementar de acordo com os requisitos pretendidos.

É de salientar que, embora alguns dos projetos possam não estar diretamente relacionados com o problema a tratar neste trabalho, o conceito de utilização de um sistema de controlo diferencial poderá ser aplicável. Estes estudos irão contribuir com ideias para resolver alguns dos problemas que poderão ser encontrados durante o desenvolvimento deste trabalho.

Desenvolvimento de uma capsula telecomandada para as operações de busca e salvamento no mar [15]

Este estudo foi desenvolvido na FEUP [15] no âmbito do projeto ICARUS para efetuar ações de busca e salvamento no mar. Este projeto tem como objetivo desenvolver uma capsula que vai ser utilizada nas operações SAR¹, em situações de desastre ou de catástrofe natural. Esta cápsula será capaz de operar de forma autónoma, por controlo remoto ou de outros modos, que serão mencionados posteriormente.

- **Requisitos funcionais do Software**

O Software instalado é composto por vários módulos que comunicam entre si. No nível mais inferior estão os módulos que interagem diretamente com os sensores e instrumentos de atuação constituindo uma camada de abstração de *hardware*. A parte superior destes dois módulos é responsável pela navegação (cálculo do tempo real da posição UCAP², velocidade e rumo) e pelo controlo (execução de manobras e outros comportamentos de nível superior). Um grupo de outros módulos específicos executa funções de supervisão, assim como a comunicação interagindo com sistemas externos, especificamente para controlo e comando. O

¹ Search And Rescue

² Unmanned Capsule

módulo de navegação processa dados dos sistemas GPS e IMU³. O sistema GPS fornece informação acerca da localização e velocidade. O sistema IMU incorpora magnetómetros, acelerómetros e giroscópios fornecendo informação sobre os três eixos de movimento (mudança de rumo ou guinadas, balanço e arfagem).

A UCAP execute missões com diferentes graus de autonomia, executando diferentes tipos de manobras em vários modos de funcionamento. Os modos de funcionamento da UCAP são:

1) Modo Autónomo que consiste numa operação autónoma com uma missão pré-programada descrita num ficheiro específico de missão.

2) Modo Externo que consiste numa operação autónoma com uma missão remotamente programada em tempo real.

3) Modo de Âncora que permite manter uma estação em execução, em que o USV⁴ mantém a sua posição compensando as flutuações causadas pelos ventos, correntes de ar ou outras influências.

4) Modo de Controlo Remoto em que a UCAP pode ser remotamente controlada pelo utilizador.

5) Modo Negligente onde a atuação da UCAP é cortada, fazendo com que flutue de acordo com as perturbações externas (ventos e correntes).

O desempenho da UCAP está diretamente relacionado com uma combinação de diferentes manobras básicas que a UCAP é capaz de fazer autonomamente. Estas manobras são:

- A navegação para um dado ponto,
- Seguir um caminho,
- Manter-se em posição estacionária e esperar por uma condição externa ou evento.

De forma a ser capaz de executar tais manobras, implementaram-se na UCAP rotinas que estão baseadas nos controladores PID para velocidade e condução num nível mais baixo e controladores de linha ou ancora num nível mais elevado.

³ Inertial Measurement Unit; é equipamento eletrónico que serve para calcular a velocidade e orientação de um veículo.

⁴ Unmanned Surface Vehicle

Estes controladores estão responsáveis por definir a ação própria dos propulsores para que o comportamento esperado seja observado e os erros de navegação sejam compensados e minimizados.

Os controladores de velocidade são responsáveis por definir a ação dos propulsores para que uma determinada velocidade de navegação seja alcançada. Os controladores de direção estão responsáveis por adaptar a ação diferencial para os propulsores para que o UCAP se dirija para o objetivo ajustando constantemente o seu comportamento.

- **Requisitos funcionais de *Hardware***

O esquema de blocos referido em baixo representa a arquitetura do sistema de controlo implementado na cápsula, que tem como unidade principal uma CPU do tipo (Pegaseus PGS800-256-2G). Este sistema computacional recebe as informações vindas dos vários sensores (GPS, IMU) e também das comunicações, e, de acordo com o *software* instalado, aciona os motores com intervenção do controlador de motores para efetuar as respetivas manobras.

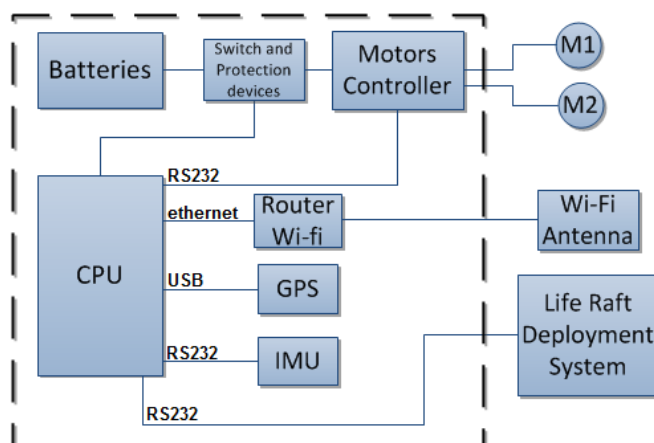


Fig. 15: arquitetura do sistema implementado na capsula

Estudo simulado de um sistema diferencial eletrónico para veículos elétricos de tração às duas rodas [16]

Este estudo que foi realizado na China [16] . Teve em consideração o facto de os veículos elétricos de tração às duas rodas (EV) terem uma configuração inovadora, na qual cada roda é impulsionada individualmente por um motor elétrico. É possível usar um diferencial eletrónico (ED) em vez de um diferencial pesado mecânico por causa do tempo de resposta rápido do motor. Uma nova aproximação de controlo ED para um sistema com dois

motores de roda EV é feita baseado no método de controlo de Lógica Difusa (*Fuzzy logic*⁵). O método *Fuzzy logic* é usado para estimar o rácio de deslize das rodas considerando a complexidade e não-linearidade do sistema. O sistema ED distribui tração e poder para cada motor de acordo com os requisitos.

A efetividade e validação do método de controlo proposto são avaliadas em Matlab/Simulink. Os resultados da simulação mostram que o novo sistema de controlo ED pode manter o rácio de deslize dentro de valores ideais, assegurando a estabilidade do veículo numa linha curva ou reta.

No entanto, o algoritmo do diferencial eletrónico é complicado devido à não-linearidade das dinâmicas do veículo. A maioria dos estudos existentes usa o modelo de *Ackermann e Jeantaud*⁶ para resolver o problema diferencial. No entanto, este modelo tem a desvantagem de ignorar a influência das forças centrífugas e centípedas quando o veículo é conduzido numa superfície curva. Em adição, a transferência de carga e o efeito dos pneus não são tidos em conta. Como tal, este estável estado de análise não é adequado quando o veículo circula em caminhos complexos com várias curvas e interligações.

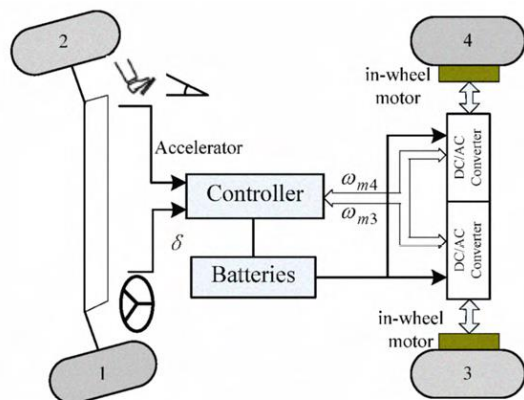


Fig. 16: Sistema de controlo e de propulsão do EV

- **Design do diferencial eletrónico**

O sistema dispõe de um motor acoplado diretamente a uma roda. Como existem dois motores no eixo traseiro da viatura, a velocidade de cada roda pode ser controlada independentemente. O controlador do diferencial elétrico é uma parte do controlador do

⁵ **Lógica fuzzy** é uma extensão da lógica booleana que admite valores lógicos intermediários entre o FALSO (0) e o VERDADEIRO (1).

⁶ *Ackermann e Jeantaud*; consiste num arranjo geométrico de ligações na direção de um carro ou outro veículo, projetado para resolver o problema do trajeto das rodas interiores e exteriores aquando de uma curva.

veículo. Está implementado num microprocessador que recebe um sinal de velocidade de cada uma das rodas e do ângulo de viragem. Após calcular a diferença de velocidade entre as duas rodas, o controlador eletrónico manda os sinais de volta para os dois controladores de motores para ajustar a velocidade da roda, independentemente. Um codificador de posição é usado para medir a posição angular do volante. Quando o ângulo de condução é igual a zero, o veículo está numa linha reta e o diferencial eletrónico não funciona. Por outro lado, se o ângulo do volante é diferente de zero ou a condição da roda esquerda é diferente da direita, esta situação indica que o veículo está a fazer uma curva. Neste caso o diferencial eletrónico começa a funcionar. A velocidade de referência comum ω_{ref} estabelece uma velocidade angular de referência correspondente à velocidade linear equivalente à posição do acelerador. Esta velocidade é definida pelo comando do pedal deste último.

Se o veículo está a virar á direita, a velocidade da roda esquerda aumenta e a velocidade da direita fica igual á velocidade de referência comum ω_{ref} . Se o veículo está a virar à esquerda, a velocidade da roda direita aumenta e a velocidade da roda esquerda é mantida à velocidade comum ω_{ref} .

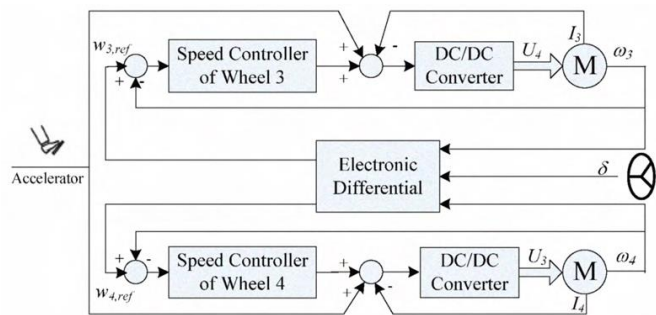


Fig. 17: Estrutura do sistema de controlo diferencial eletrónico

A figura 18 apresenta o princípio funcional do controlo de sistema proposto para o diferencial elétrico. Um método de controlo de feedback de *loop* duplo é adotado para o controlo do motor. O controlo de *loop* interior controla cada corrente de cada motor e os controlos de *loop* exteriores controlam a tração do motor para ajustar a velocidade de cada motor. Um microprocessador e vários cartões de interface eletrónica foram usados para implementar o diferencial eletrónico. Um sensor de voltagem e de corrente é usado para cada motor, providenciados os sinais necessários para a estima da velocidade de rotor.

Desenvolvimento de algoritmos de navegação com *WayPoints* para os helicópteros Quadrotor [17]

Este projeto realizado na Índia [17] teve como objetivo desenvolver um algoritmo para navegação autónoma com *WayPoints* usando um sistema de *Global Positioning System* (GPS) e um controlador de bordo do tipo *Programmable Interrupt Controller* (PIC). As coordenadas dos *WayPoints* para um rumo pré-determinado são introduzidas no microcontrolador. O algoritmo em questão vai calcular a distância entre a posição atual obtida pelo GPS e o primeiro *WayPoint* usando a fórmula de *haversine*⁷ e o azimuth ao *WayPoint* a partir da nossa posição atual referente ao Norte Geográfico. Igualmente o mesmo pode ser calculado para sucessivos *WayPoints*. Através dos valores calculados o sinal *Pulse Width Modulation* (PWM) é gerado pelo PIC para controlar a atitude do *QuadRotor*. Esta função é utilizada em aplicações, tais como; vigilância de segurança autónoma, sistema supressor de fogo, e um veículo topográfico.



Fig. 20: Trajetória composta por vários WayPoints



Fig. 19: O Quadrotor a seguir a sua trajetória

- **Requisitos funcionais de Hardware**

Neste projeto o GPS é usado como um sensor Input que fornece as coordenadas geográficas (latitude e longitude) da posição atual ao controlador. Os dados GPS são processados pelo PIC através de um algoritmo de controlo. O ponto de destino (latitude e longitude) estão no sistema pois fazem parte do plano de voo. O controlador calcula então a distância entre a posição inicial e o *WayPoint* (WP1) de destino. O azimuth ao *WayPoint* de destino a partir da posição atual, é também calculado pelo algoritmo com referência ao Norte Geográfico. Foi estudada em pormenor a funcionalidade do GPS e a interface do GPS com o PIC.

⁷A fórmula de *Haversine* é uma equação usada em navegação, fornecendo distâncias entre dois pontos (d) de uma esfera de raio R, a partir de suas latitudes (ϕ) e longitudes (λ).

A equação é então dada por: $hav(\frac{d}{R}) = hav(\Delta\phi) + \cos \phi_1 \cos \phi_2 \text{ hav}(\Delta\lambda)$

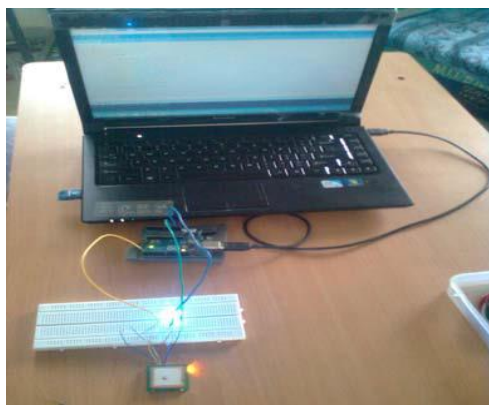


Fig. 21: GPS ligado ao microcontrolador

Desenvolvimento de controladores de navegação de baixo custo para os veículos autónomos terrestres [18]

Esta investigação feita nos EUA [18], aponta no desenho de um rentável sistema de navegação de baixo custo e no desenvolvimento da capacidade de minimizar o erro na estima da navegação de um (UGV) autónomo, por forma a otimizar o comportamento relativo à deslocação de um UGV. Para completar este objetivo foram tomados os seguintes passos:

- Elaborar um processo de investigação para explorar desafios e técnicas dos sistemas de navegação de um UGV autónomo;
- Desenhar um UGV usando componentes de controlo e sensoriais comerciais (COTS);
- Implementar um filtro Kalman⁸ para otimizar a estima do rumo do UGV;
- Integrar o subsistema de controlo e de sensores do UGV usando um Freescale 9S12 microcontroller⁹;
- Comparar a performance do UGV usando COTS com a do UGV usando Freescale 9S12 microcontroller;

O *design* de um UGV telecomandado é muito simples pois apenas requer capacidades moderadas de computação que têm que estar presentes no UGV. No entanto, no caso de um UGV autónomo, não existe a mínima interação humana para sua operação. Um UGV autónomo tem as habilidades de:

- Ganhar conhecimento do terreno em que se situa;

⁸ O filtro de *Kalman* é um método matemático criado por *Rudolf Kalman* que produz estimativas dos valores reais de grandezas, medidas e valores associados.

⁹ *Freescale 9S12 microcontroller* é microcontrolador da família *Freescale Semiconductor*.

- Operar sem intervenção humana;
- Navegar para uma localização específica sem ajuda humana;
- Controlar o seu movimento dependendo do terreno circundante;
- Detetar e evitar obstáculos;
- E aprender novas habilidades sem qualquer assistência exterior.

Um UGV autónomo requer moderada/alta computação que garanta as habilidades referidas supra. A navegação de um UGV necessita sensores internos de movimento e em alguns casos também precisa de assistência externa como satélites GPS ou antenas. Um UGV que use sensores de movimento para prever qual a posição, velocidade e orientação a tomar são compostos por um Sistema INS (Inertial Navigation System). Um INS usa um computador e dispositivos sensoriais de movimento como acelerómetros, giroscópios, etc. Com um conjunto deste tipo de sensores, o sistema pode mudar os seus parâmetros de posição, uma vez que está continuamente a ser fornecida informação de posição. Um INS de baixo custo, normalmente apresenta diversos problemas, tais como erros de escala e erros de derivação. O erro de derivação é causado por acumulação de pequenos erros de medição, tanto de velocidade linear, como de velocidade angular. Estes pequenos erros quando somados originam grandes erros na estima de posição, velocidade e orientação do UGV.

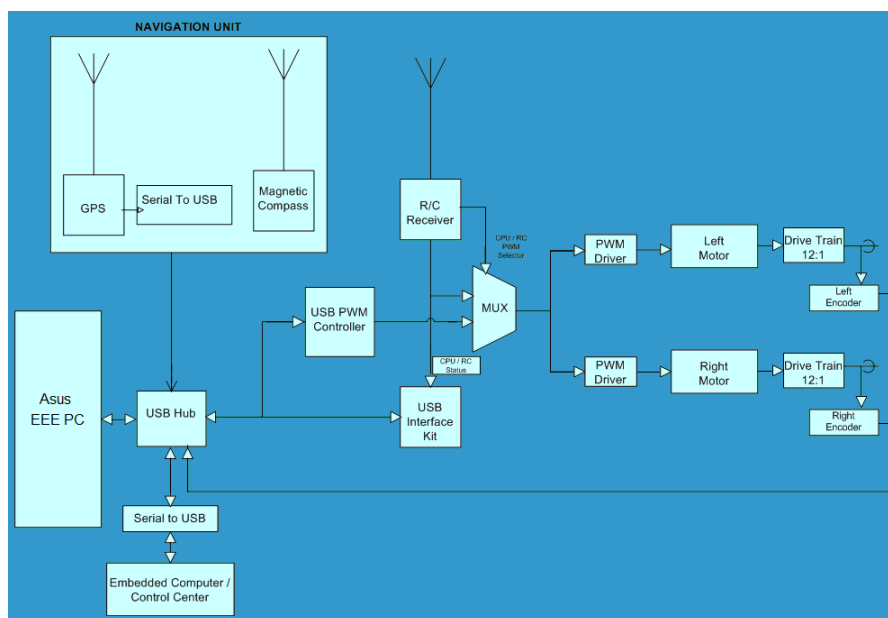


Fig. 22: Subsistema eletrónico do UGV

O módulo de GPS suporta uma interface RS-232¹⁰, mas necessita de um conversor USB para a interface do GPS com o USB. O EEE PC (*Notebook Computer*) atua como um ponto primário de controlo para navegação autónoma do UGV, e executa as seguintes funções:

- Recebe comandos do computador principal e executa-os;
- Combina informação de localização do GPS e informação de proa da bússola para atualizar o computador mãe sobre a sua localização e orientação;
- Recebe informações do codificador de eixo das rodas de ambos os lados para calcular a distância viajada pelo UGV;
- Monitoriza o estado das baterias do motor e do computador;
- Controla o movimento do motor usando o controlador de motores *Phidget* para executar as manobras;
- Compila a informação do estado de operação do robô pelo controlador rádio para executar comandos quando o UGV estiver em modo autónomo;
- Ignora os comandos do UGV recebidos pelo computador mãe quando o UGV em modo manual.

Por forma a implementar recursos tais como gerenciamento remoto de energia, controlo manual e capacidade de mudar o modo de operação do UGV entre manual/autónomo, a placa de circuito é desenhada por forma a multiplicar os sinais por controlo computacional ou rádio baseando-se no modo de operação do UGV.

2.8- Síntese dos estudos referidos

Foram referidos anteriormente vários sistemas de controlo com diversas implementações de forma detalhada e pormenorizada dos seus requisitos funcionais. Essas implementações apresentam soluções que podem ser usadas no desenvolvimento de uma solução de baixo custo e que permita alcançar os objetivos propostos neste trabalho de investigação.

Relativamente aos microcontroladores, cada solução estudada utiliza um equipamento diferente. No projeto da FEUP (primeiro estudo acima referido) [15] foi usado o PC/104 que é muito mais do que um microcontrolador a bordo do UCAP e que tem uma grande capacidade de memória e de processamento. Pode suportar o *software* que requer

¹⁰ **RS-232** (também conhecido por **EIA RS-232C** ou **V.24**) é um padrão de protocolo para troca serial de dados binários

complexidade de cálculo e contém vários modos de funcionamento que são necessários para cumprir com os objetivos traçados. No projeto de desenvolvimento do *Quadcopter* [17] foi usado o Atmega-328P que carrega o *software* responsável por fazer a navegação autónoma com *WayPoints*. O Atmega-328P é um PIC que pode ser encontrado nas placas de desenvolvimento Arduíno disponíveis no CINAV. No resto dos projetos dos estudos acima referidos foram usados o Arduíno para fazer diversas tarefas. Este sistema de desenvolvimento é baseado em microcontroladores PIC que têm uma boa relação custo/capacidade/facilidade de uso. Como tal, um Arduíno apresenta ser uma solução viável para este projeto uma vez que é usado em soluções semelhantes.

Relativamente aos sensores, para se conseguir fazer a navegação de uma embarcação foi utilizado um GPS e um sistema IMU no projeto do primeiro estudo acima referido [15]. Estes dois equipamentos são complementares pois o GPS fornece-nos informações acerca da localização da embarcação, enquanto o IMU, como é um sistema que incorpora magnetómetros, acelerómetros e giroscópios, fornece-nos informação sobre os três eixos de movimento (mudança de rumo ou guinadas e balanço bombordo-estibordo e balanço proa-popa). Assim, para este projeto, pretende-se também usar um módulo GPS para indicar a posição, sendo complementado por um módulo que incorpora uma bússola magnética. A bússola magnética tem como objetivo indicar sempre a proa da embarcação.

Relativamente aos requisitos de *software*, no projeto da FEUP [15] foram utilizados diversos modos de funcionamento que são extremamente necessários para o bom cumprimento da missão de busca e salvamento no mar que é objetivo principal desse projeto e deste também. No entanto pode-se aproveitar muito bem a forma como foram implementados foram usados esses modos de funcionamento e a maneira como interagem entre eles.

2.9- Especificação dos requisitos do sistema

Após as conclusões tiradas dos vários estudos referidos em cima, e tendo em conta o fato de haver uma variedade considerável de arquiteturas e sistemas possíveis para serem adaptados ao nosso problema, iremos aproveitar estes mesmos para explorar os requisitos que estão ao nosso dispor CINAV. Com isso tudo darei nesta subsecção uma ideia ampla sobre aquele que irei precisar para resolver o problema em termos de requisitos de *software* e de *hardware*.

Requisitos de *hardware*

O sistema de controlo que pretendemos implementar é composto por uma peça central que é um microcontrolador, os conjuntos de sensores que lhe passarão as informações necessárias, e dois motores que cumprem as decisões tomadas pelo microcontrolador de acordo com aquele que foi recebido dos sensores e de forma a cumprir o seu objetivo. Esses sensores disponibilizam informações como, posição GPS, Rumo, Velocidade, que serão usadas pelo microcontrolador para interagir com o Software de um lado e realizar o que é pedido de outro lado.

Em termos de sistema de controlo podemos dividi-lo em quatro componentes:

- Sistema Microcontrolador
- Sistema propulsor (motores elétricos *TORQEEDO*)
- Sensores
- Sistema do comando remoto (TX e RX)

- **Sistema de microcontrolador**

O microcontrolador escolhido vai ser responsável por correr o *software* que irá controlar os motores e ler a informação dos sensores, para conseguir tomar decisões. Escolhemos para este projeto o Arduíno Mega 2560. Devido ao facto de já ter sido aplicado em projetos anteriores na Escola Naval e noutros projetos fora da EN. No Anexo do Arduíno é descrito um pouco do conceito do Arduíno em termos de Hardware.

Escolhemos o Mega 2560 pois é o modelo com maior capacidade de Memória, de processamento e com mais entradas e saídas. O anexo do Arduíno refere o tipo de Arduíno que há e descreve especificamente o Arduíno Mega 2560.

- **Sistema propulsor**

O sistema propulsor ou os motores são necessários para atuar na nossa embarcação e fazê-la movimentar e fazer as respetivas manobras para cumprir com os objetivos.

Em avanço à existência de especificações técnicas da embarcação no que se refere ao seu sistema propulsor, utilizar-se-ão como meios de experimentação, os motores propulsores elétricos e uma embarcação existentes no CINAV. Esses propulsores elétricos são dois motores

Torqueado. Sobre estes motores já foi estudado o seu comando remoto e foi decodificada a linguagem usada entre a alavanca, o sensor magnético e o microprocessador do respetivo motor pelo Segundo Tenente EN-AEL Jorge de Jesus (dissertação Jesus 2013). Em desenvolvimento também encontra-se a modelação de hélices disponíveis para o mesmo motor por parte dos camaradas (CAD EN-MEC Cardoso da Silva e CAD EN-MEC Mártires Paulino) por forma a melhora-los em termos de eficiência para as suas mais diversas aplicações e exigências operacionais.

- **Sensores**

Como tal para conseguirmos controlar a nossa embarcação usando os dois motores acima referidos e o *Arduíno Mega 2560* como microcontrolador, precisamos também de um GPS e uma bússola para nos fornecer as informações necessárias como a posição, rumo, velocidade etc.

A bússola será importante porque precisamos de saber a proa da nossa embarcação para conseguirmos fazer as guinadas necessárias de modo a seguir o rumo pretendido. Portanto a nossa embarcação precisa de uma bússola para fornecer esta informação. O GPS também nos fornece esta informação, mas só quando estamos a andar para frente, tendo uma baixa precisão. Entretanto, não é possível usar a informação da proa vinda do GPS porque quando a embarcação está a virar não está a andar para frente.

- **Sistema do comando remoto (TX e RX)**

Este sistema será responsável por transmitir as ordens do operador para o microcontrolador em termos de mudar de um modo de funcionamento para outro. Também irá permitir fazer as manobras, quando se encontra no modo de controlo remoto.

Requisitos de *software*

O *software* da camada mais baixa tem de ser capaz de ler os valores dos sensores e de definir os estados dos motores elétricos e controlá-los de forma eficaz com base nos dados obtidos pelo Software das camadas acima. Em suma, o *software* tem que ser capaz de ler dados de uma bússola digital, de um GPS, bem como qualquer outro sensor. Também tem que ser capaz de interpretar os sinais recebidos pelo recetor que por sua vez recebeu do transmissor e interagir com as várias funcionalidades e modos que ele contém por forma a realizar o que o operador pretende.

Modos de funcionamento

- **Modo Autónomo:** que consiste numa operação autónoma com uma missão pré-programada descrita num ficheiro específico de missão.
- **Modo de Controlo Remoto:** Em que a embarcação pode ser remotamente controlada pelo utilizador.

O nosso algoritmo também terá de ser capaz de permanecer constantemente em escuta, e conseguir trocar o seu modo de funcionamento através da mudança de uma parte de código para outra, executando-a de forma eficiente. Por fim, terá de conseguir fazer a comunicação entre os diversos modos, por forma a realizar outro tipo de tarefa.

Estes são os pontos que se vão desenvolver. O que se pretende nesta primeira fase do projeto é que a nossa embarcação consiga trabalhar, pelo menos, em dois modos, os quais não serão suficientes para este tipo de projetos e para o bom cumprimento da missão SAR. Assim, será necessário fazê-la evoluir para trabalhar noutros modos tais como, modo âncora e modo negligente (FEUP [15]).

Neste capítulo será descrito o trabalho desenvolvido. Este capítulo divide-se em duas partes, Hardware usado no sistema de comando e controlo, e o Software ou o algoritmo usado para o sistema de comando e controlo. Na parte de Hardware são descritos todos os componentes utilizados para o desenvolvimento do projeto. No subcapítulo de Software é apresentado toda a arquitetura do sistema, bem como o funcionamento do sistema de controlo automático.

3.1. Hardware usado no sistema de comando e controlo

Nesta secção serão apresentados os vários módulos que compõem a Arquitetura proposta para o sistema de comando e controlo de dois motores elétricos. O sistema é composto por um módulo de navegação, onde está incluído um GPS e uma bússola magnética, e um módulo de controlo de motores compostos por um sistema de desenvolvimento Arduino e uma placa de interface para ligar o Arduino ao comando do motor como demonstra a figura em baixo. Os motores elétricos usados neste projeto são da marca *Torqueedo*, modelo 2.0R.

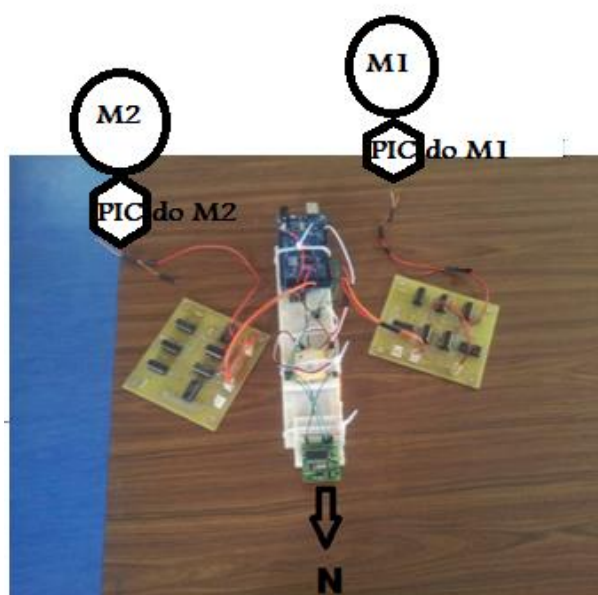


Fig. 23: Diagrama do Hardware que será utilizado

3.1.1. Motor elétrico

Torqueedo

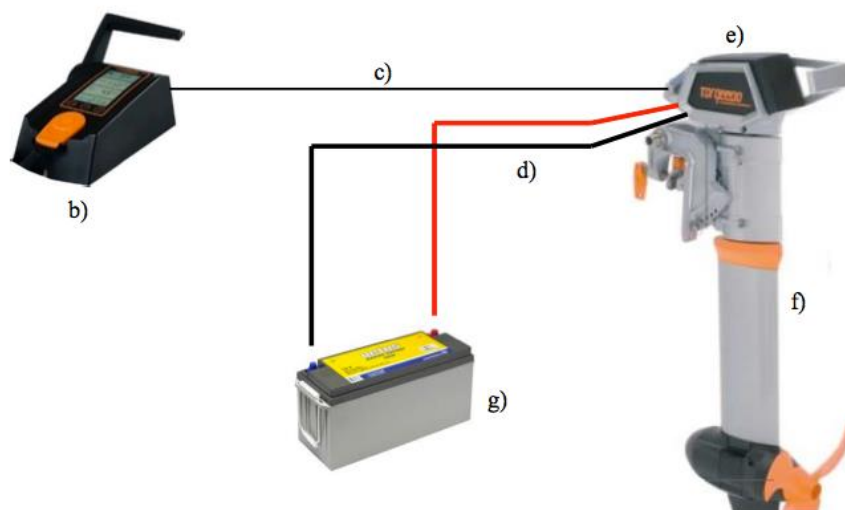
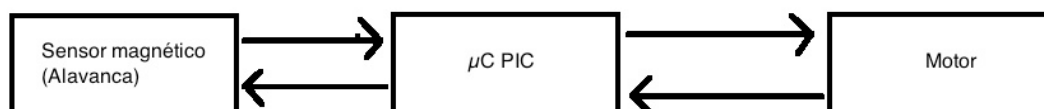


Fig. 24: b)-controlador de velocidade, c)-cabo de ligação entre o controlador de velocidade e o motor, d)-ligação do motor s baterias, e)-modulo GPS, f)-motor, g)-baterias (Jesus,2013).

No esquema de ligações apresentado, o controlo de velocidade é feito no modulo de controlo indicado em b). O sistema de substituição da alavanca de controlo de velocidade pretende substituir a alavanca que faz parte do módulo b), por um sistema desenvolvido no âmbito desta tese.

O sistema de controlo dos motores *Torqueedo* é ilustrado na fig.25. A alavanca de controlo de velocidade dispõe de um magneto que varia as linhas de força do campo magnético conforme a posição da alavanca. Junto da alavanca existe um sensor magnético (AS5045) que deteta as alterações do campo magnético e codifica numa sequência binária. Este sensor comunica com um microcontrolador PIC que por sua vez aciona o motor como demonstra a fig.2.



Outra Fig. 25: Esquema geral da comunicação da alavanca com o informação importante

presente no manual do *Torqueedo* é o facto das palavras enviadas variarem linearmente, consoante a posição angular da alavanca. Esta informação será muito útil no envio da palavra binária, que corresponde à percentagem de força dos motores, para o PIC do motor. Esta situação vai permitir que a velocidade a aplicar aos motores seja determinada de uma forma simplificada e rápida.

Depois de conhecidas as principais características do sistema de controlo do motor, foi inicialmente tentada a sua substituição integral. Para tal seria necessário programar um novo

microcontrolador capaz de o substituir. Somente conhecendo todos os protocolos de comunicação entre o motor e o PIC seria possível programar esse microcontrolador.

Os dados recolhidos desta comunicação não foram suficientes para conseguir identificar o significado de todas as tramas enviadas pelo motor. Também não foi possível distinguir os dados oriundos do PIC e do motor. (Jesus, 2013).

Assim, a decisão foi a da replicação do sensor magnético através do Arduino.

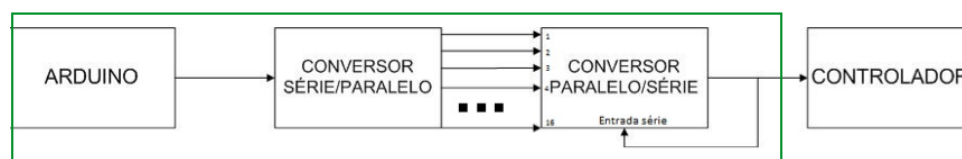
3.1.2. Placa de desenvolvimento Arduino

Visto que o PIC decide quando a palavra DO¹¹ é recebida do sensor magnético, para garantir o sincronismo no envio de dados do *Arduino* para o PIC foi necessário desenvolver um circuito que pudesse armazenar os 16 bits gerados pelo *Arduino* e sincronizar o seu envio para o PIC, sempre que este o assim pedir.

Para além disso, durante intervalos de tempo tão pequenos, não é desejável enviar sempre uma palavra diferente, como tal, pensou-se fazer um *buffer* de armazenamento temporário que disponibilize ao PIC de controlo do motor uma palavra binária de velocidade sempre que necessário.

O *buffer* consiste em dois circuitos do tipo *shift-register*. O *Arduino* quando pretende enviar um valor de velocidade diferente para o motor, envia uma sequência binária em série para o primeiro *shift-register*. Este circuito vai atuar como memória intermédia temporária. Quando o PIC do controlador do motor pretende ler a posição da alavanca (ou posição simulada), envia um bit de controlo a indicar esta ação. O valor guardado no primeiro *shift-register* é passado em paralelo para o segundo *shift-register*, ficando o sistema sempre com uma cópia guardada em memória. O conteúdo do segundo *shift-register* é enviado em série para o PIC. Durante esta ação a informação é perdida. Optou-se por não ligar a saída do *shift-register* à entrada por falta de sincronismo no número de bits de *clock* necessários para repor a palavra original.

Com isto, tem-se a transmissão geral dos dados do *Arduino* para o controlador do motor.



¹¹ Digital Output é uma palavra binária de 16 bits com informação da velocidade

Fig. 26: Sistema implementado substituto do sensor magnético

3.1.3.Cálculo da curva de velocidade dos motores

Para simular as palavras a serem enviadas, teremos de saber primeiro como são geradas. Já sabemos que cada palavra corresponde a uma determinada velocidade e que são linearmente proporcionais à percentagem de potência imposta. Portanto, com a ajuda de um osciloscópio, identificámos algumas palavras, nomeadamente: máxima força a vante; máxima força a ré; motor parado e as duas palavras que correspondem ao limite entre o motor estar parado e começar a andar (nos dois sentidos). A estas palavras “limite” fez-se corresponder a potência de 1%.

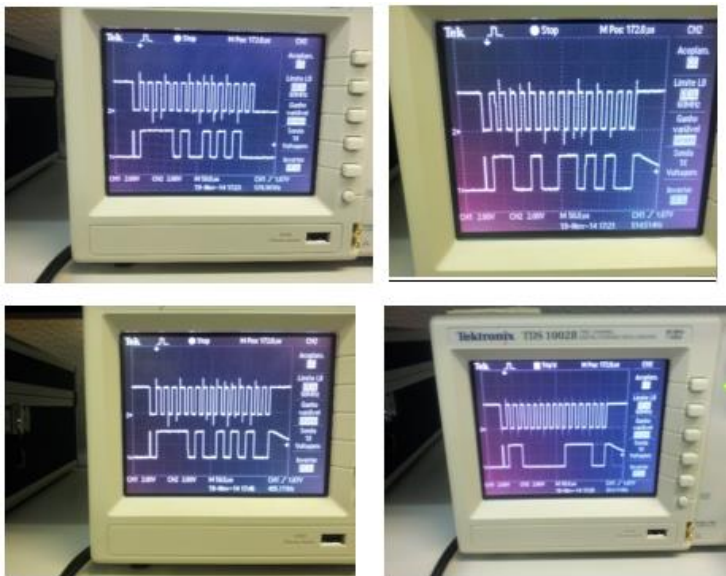


Fig. 27: A sequência binária DO com várias posições da alavanca

A tabela abaixo mostra a correspondência das palavras em binário e em decimal com as várias percentagens de forças do motor.

	100% a vante	100% a ré	Motor parado	1% a vante	1% a ré
Decimal	3289	619	4066	3879	50
Binário	110011011001	1001101011	111111100010	111100100111	110010

Com estes dados, calculou-se as duas equações de retas correspondentes (para a ré e para avante): (Note-se que as percentagens de força a ré, equivalem valores de \underline{x} negativos).

$$w_f = \frac{3289 - 3873}{100 - 1} \approx -5.9 \rightarrow F(x) = -5.9x + S_f$$

$$F(100) = -5.9 \times 100 + S_f = 3289 \leftrightarrow S_f = 3289 + 590 = 3879$$

$$F(x) = -5.9x + 3879, \quad x \in]0, 100]$$

$$w_R = \frac{619 - 50}{1 - 100} \approx -5.75 \rightarrow G(x) = -5.75x + S_R$$

$$G(-100) = -5.75 \times (-100) + S_R = 619 \leftrightarrow S_f = 619 - 575 = 44$$

$$G(x) = -5.75x + 44, \quad x \in [-100, 0[$$

Estas duas funções estão representadas no seguinte gráfico:

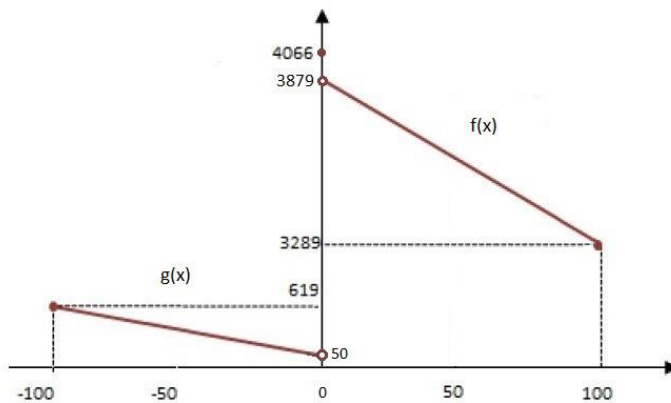


Fig. 28: Curva de velocidade do motor

Quando o utilizador decide a velocidade desejada, o programa irá calcular a palavra correspondente, convertê-la para um número binário de 12 bits, e acrescentar os 4 bits constantes, 1001, no final.

-Pontos principais do algoritmo desenvolvido em Arduíno para mandar a velocidade desejada para os motores

-Acrescentar os bits constantes, 1001

```
velocidade = 4066;
velocidade = velocidade << 4;
velocidade = velocidade | 9;
```

4066 (DEC) = 111111100010 (BIN)

Ao executar $4066 \ll 4$, serão acrescentados 4 bits de valor 0 à direita do valor binário correspondente a 4066, pois os bits são transladados 4 casas para a esquerda.

$4066 \ll 4$ (DEC) = 111111100010**0000** (BIN)

A operação seguinte é chamada de *bitwise OR* e atua como um lógico *OR*, para cada bit independentemente, entre os dois operandos em questão.

$4066 \mid 9$ (DEC) = 111111100010**0000** + **1001** = 111111100010**1001**

Envio da sequência de bits

```
shiftOut(dataPin,clockPin,MSBFIRST,(data>>8));  
shiftOut(dataPin,clockPin,MSBFIRST,data);
```

O objetivo é transmitir uma palavra de 16 bits em série, enviando em primeiro lugar o bit mais significativo. A função *shiftout* do Arduíno é limitada à transmissão de apenas 8 bits de cada vez, e como tal, em primeiro lugar serão enviados os 8 bits mais significativos, utilizando um operador semelhante ao já demonstrado no tópico anterior. O parâmetro “MSBFIRST” significa “Most Significant Bits First”, o que faz com que seja enviado a sequência do bit mais significativo até ao menos significativo.

É de salientar que se fez dois processos do mesmo acima uma para cada motor.

3.1.4. Placa de interface

Para implementar o bloco conversor série/paralelo de 16 bits foram utilizados dois circuitos integrados SN74LS164N de 8 bits em série.

Para o bloco conversor paralelo/série de 16 bits foram utilizados dois circuitos integrados DM74LS165N de 8 bits em série.

Lógica adicional tem de ser implementada, pois existem certas restrições de quando o *Arduíno* pode transmitir os dados, e de quando os integrados os podem receber e reenviar.

Para as portas AND foi utilizado um circuito integrado 7408N, para os inversores foi utilizado o SN74LS04N e para o flip-flop D o SN74LS74AN.

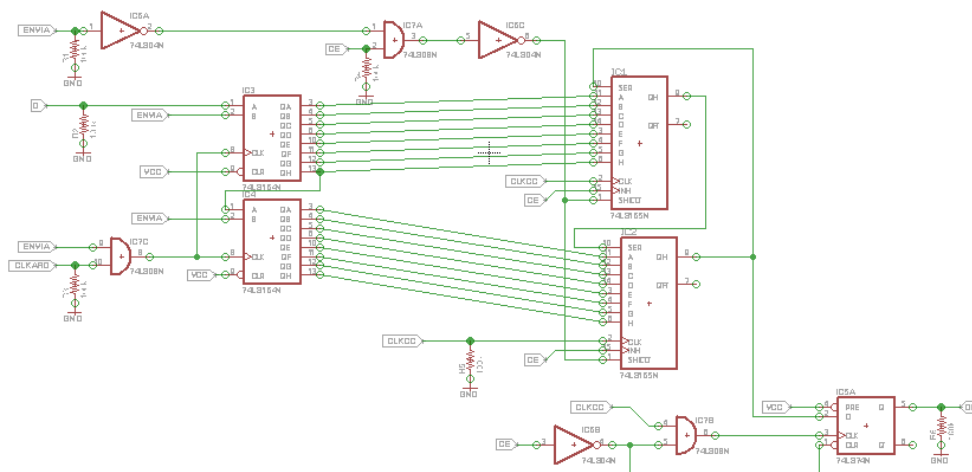


Fig. 29: Esquemático do circuito lógico

Em cada linha de comunicação entre o circuito acima descrito e o *Arduino* ou o controlador, colocou-se uma resistência *pull-down* para reforçar o valor lógico pretendido, caso alguma entrada seja desconectada a entrada terá valor lógico 0.

No esquemático final foram também colocados dois condensadores de 100nF em paralelo entre o GND e o VCC dos circuitos integrados, para filtragem de ruído.

Para teste do circuito utilizámos o *Arduino* para simular o envio de palavras, e os inputs/outputs do controlador do motor.

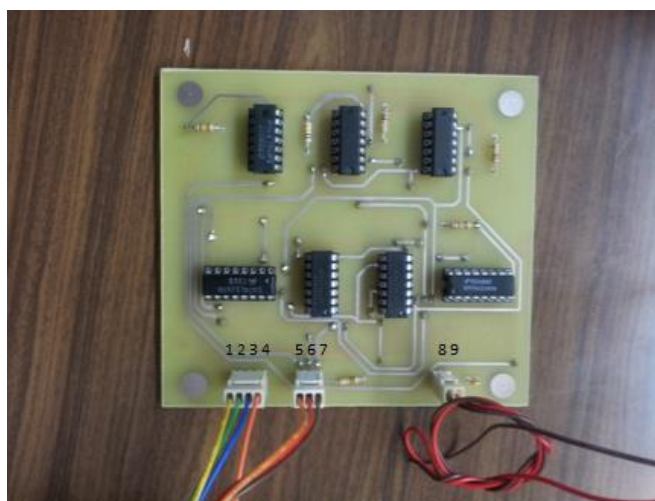


Fig. 30: Circuito impresso final

Legenda de INPUTS E OUTPUTS:

1. **CLK:** Clock do controlador.
2. **DO:** Saída da sequência binária.
3. **CE:** Chip Select do controlador.

4. **GND:** Ground.
5. **CLK Arduino:** Clock do Arduino.
6. **ENVIA:** Sinal de controlo do Arduino.
7. **D:** Entrada da sequência binária.
8. **Vcc:** 5V.
9. **GND:** Ground.

O estudo sobre o funcionamento do controlo do motor elétrico Torqueado foi iniciado pelo 2TEN EN-AEL Jorge Jesus (2013) e complementado com o trabalho desenvolvido na Escola Naval pelos alunos Afonso Mendes Ferreira e Guilherme Rodrigues Gaspar, da Faculdade de Ciências e Tecnologia, da Universidade de Nova de Lisboa.

3.1.5. Sensores

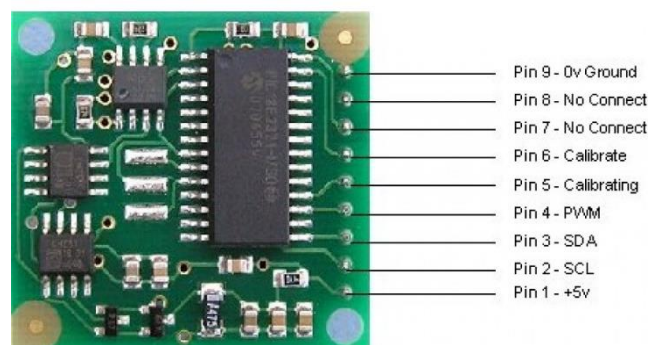
Os sensores que serão utilizados neste projeto serão o GPS e a agulha magnética que serão descritos a seguir:

3.1.4.1. Agulha magnética

A bússola será importante porque precisamos de saber a proa da nossa embarcação para conseguirmos fazer as guinadas necessárias de modo a seguir o rumo pretendido. Portanto a nossa embarcação precisa de uma bússola para fornecer esta informação. O GPS também nos fornece esta informação, mas só quando estamos em movimento, tendo uma baixa precisão.

Tendo a distância e o rumo a partir do GPS, agora precisa-se saber qual é a proa da embarcação para conseguir fazer guinadas eficientes e ter a proa da embarcação sempre em correspondência com o rumo dela. Foi usado o modelo COMPS03. O COMPS03 é bastante conhecido e utilizado no mundo da robótica, devido não só ao seu baixo custo, como a sua boa precisão, e fácil utilização. A figura em baixo mostra o significado de cada pino. Sendo que o Vcc (normalmente +5V); GND (massa comum entre os equipamentos, muito importante para que quando os dispositivos queiram dizer zero, o consigam dizer de forma perceptível para todas as outras máquinas);

SDA (Serial Data)
SCL (Serial Clock).



Line);

Fig. 31: Bússola modelo COMPS03

A comunicação entre a bússola e o Arduino vai ser feita através do protocolo I2C [x], pois este permite uma boa modularização das comunicações, baixando o tempo e custo de desenvolvimento de dispositivos, assim como uma grande flexibilidade no funcionamento, consumindo pouca corrente, e sendo bastante imune a ruídos.

A figura em baixo mostra a montagem para ler a proa da bússola eletrónica usando Arduino. Primeiro passo, ligar GND e Vcc do Arduino ao sensor, (podemos optar pelos 5V ou pelos 3,3V se quisermos meter num bus que já tem outros componentes a 3,3V, pois este sensor adapta-se a qualquer uma das tensões), assim como o SCL e o SDA aos pins 21 e 20 do Arduino Mega2560 respetivamente. Utilizamos os pinos 20 e 21, porque o microcontrolador o *ATmega2560* de que é feito o *Arduino Mega2560*, só implementa o protocolo I2C nestes pinos, e a biblioteca *wire.h* que vamos utilizar não implementa I2C por *software*, mas sim faz a bridge para a implementação do *ATmega*.

Com isto podemos ou não utilizar umas resistências de *pull up* pois é aconselhado neste protocolo.

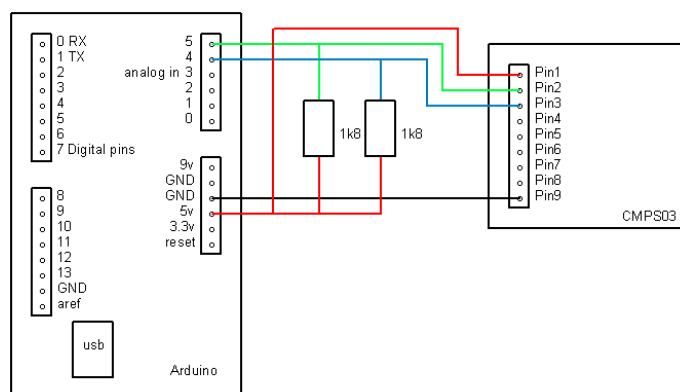


Fig. 32: Montagem para ler a bússola

Temos ter atenção que este sensor deve de ser colocada o mais afastado possível de interferências magnéticas, tal como motores, e outras fontes de campos, a fim de se poder obter valores corretos.

3.1.4.2. GPS

A primeira meta traçada foi conseguir fazer ler e registrar todos os dados que o GPS debitava, o sensor que usei foi o GPS *GlobalSat EM 406*. Antes de se conseguir fazer a montagem para ler os dados (Fig.15), foi necessário preparar o sensor e saber o que é que cada pino faz (Fig.14).

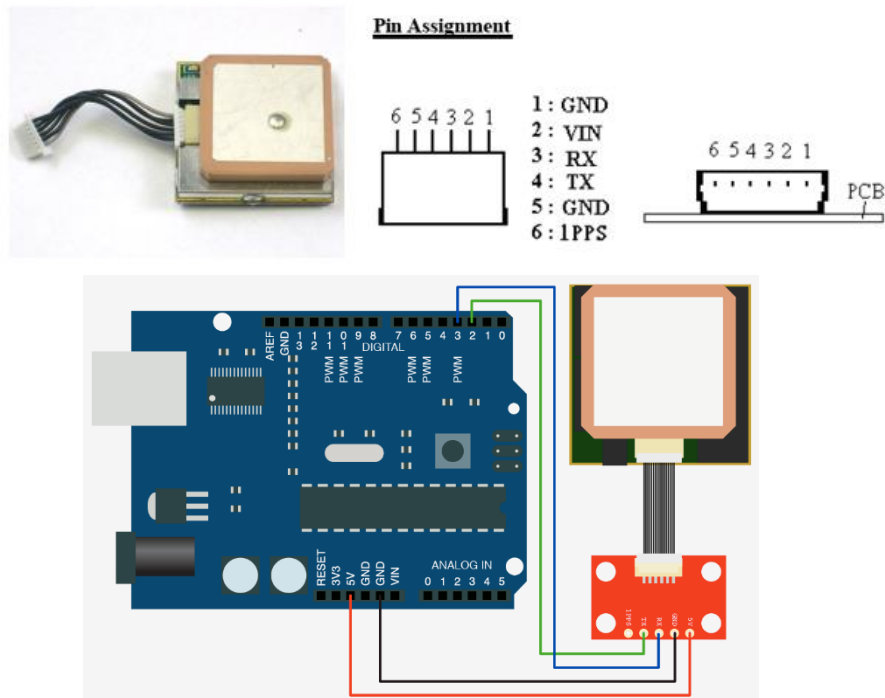


Fig. 34: Montagem para ler o GPS

O que conseguimos obter do GPS são várias frases chamadas NMEA¹² frases Fig.34 que contém muita informação que pode ser aproveitada em varias aplicações Fig.16, sendo que no nosso projeto será preciso só a informação que refere à posição geográfica da nossa embarcação. As frases do GPS que podemos usar são GPRMC e GPGGA sendo que estas contém tudo que precisamos, e também é mais fácil para retirar a latitude e a longitude da nossa posição geográfica. Os dados são separados com vírgulas para facilitar a leitura e a análise para o microcontrolador.

A importância dos pontos GPS reflete no facto do conhecimento da posição da embarcação em relação ao *WayPoint*, nomeadamente a distância. Vai ser referida de seguida

como se pode utilizar as latitudes e as longitudes que o GPS fornece para calcular a distância até ao próximo *WayPoint*.

```
$GPRMC,235316.000,A,4003.9040,N,10512.5792,W,0.09,144.75,141112,,*19
$GPGGA,235317.000,4003.9039,N,10512.5793,W,1,08,1.6,1577.9,M,-20.7,M,0000*5F
$GPGSA,A,3,22,18,21,06,03,09,24,15,,,,,2.5,1.6,1.9*3E
```

Por exemplo, A frase GPGGA contém o seguinte :

- Time: 235317.000 is 23:53 and 17.000 seconds in Greenwich mean time
- Longitude: 4003.9040,N is latitude in degrees.decimal minutes, north
- Latitude: 10512.5792,W is longitude in degrees.decimal minutes, west
- Number of satellites seen: 08
- Altitude: 1577 meters

Fig. 35: Demonstração das frases lidas pelo GPS e a informação que contém

3.1.6. Comando remoto

Foi utilizada a função *PulseIn* para ler os sinais *PWM* dos canais enviados do transmissor para o recetor. Esta função Lê um pulso (tanto HIGH como LOW) em um pino. Por exemplo, se valor for HIGH, *pulseIn()* espera que o pino vá para HIGH, inicia a cronometragem, e então espera que o pino vá para LOW e para a cronometragem. Retorna a duração do pulso em microssegundos. Para variar a velocidade dos motores é necessário alterar o “valor”, alterando assim a largura de pulso e consequentemente o Duty Cycle¹³. A Figura 36 apresenta vários Duty Cycle’s que o recetor possa receber do transmissor.

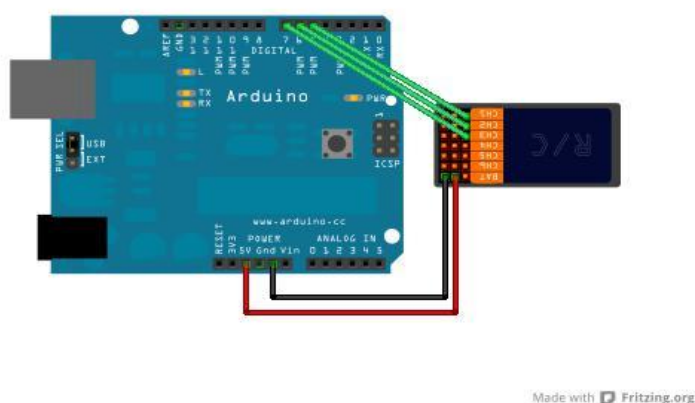


Fig. 37: montagem para ler os sinais dos canais do recetor

¹³ Duty Cycle é a proporção de tempo durante o qual um componente, dispositivo ou sistema está em operação, neste caso em 5V.

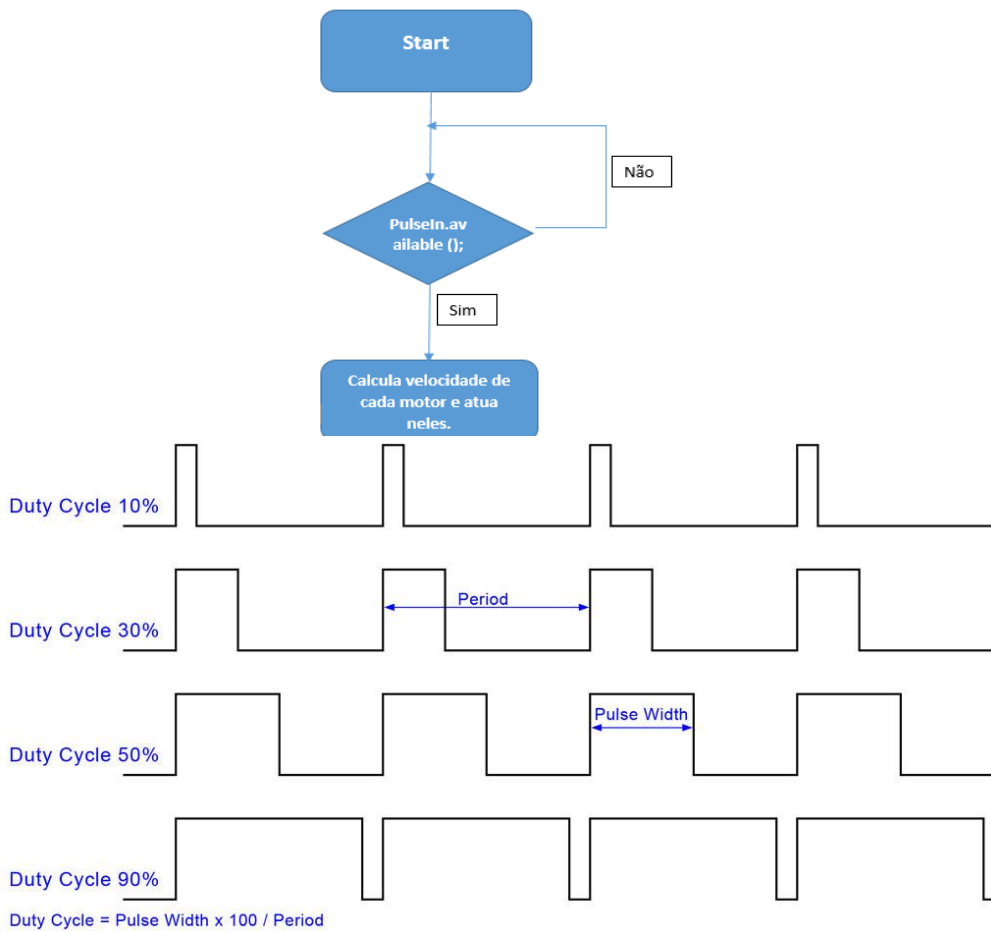


Fig. 39: Pulse Width Modulation e Duty Cycle

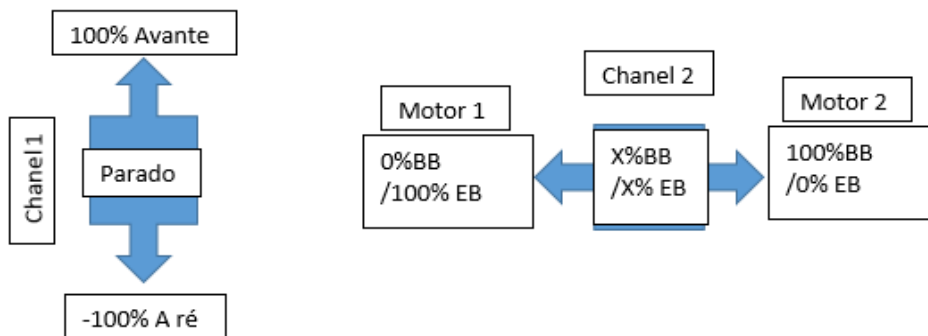


Fig. 40:Princípio de funcionamento do telecomando

Como mostra a fig.38, para conseguirmos controlar a velocidade dos motores com um telecomando TX/RX, precisa-se de pelo menos 2 canais, sendo um deles responsável por fazer os motores andarem para vante ou para ré, e o outro responsável pela parte das guinadas, que é feita reduzindo a velocidade do motor do lado para onde queremos guinar, assim nunca será possível efetuar guinadas rápidas. Essa redução da velocidade será feita da seguinte forma:

Se o operador quiser guinar para direita irá puxar o Joystick para direita e depois consoante a intensidade da guinda ou (a diferença entre a posição do Joystick no meio até onde o operador parar de puxar). Esta intensidade de guinada ou taxa de viragem será calculado e convertida em percentagem de força (de 0 a 100 %).

Retirar está percentagem da percentagem total do motor da mesma borda e assim reduzimos a velocidade do motor do lado para onde queremos guinar.

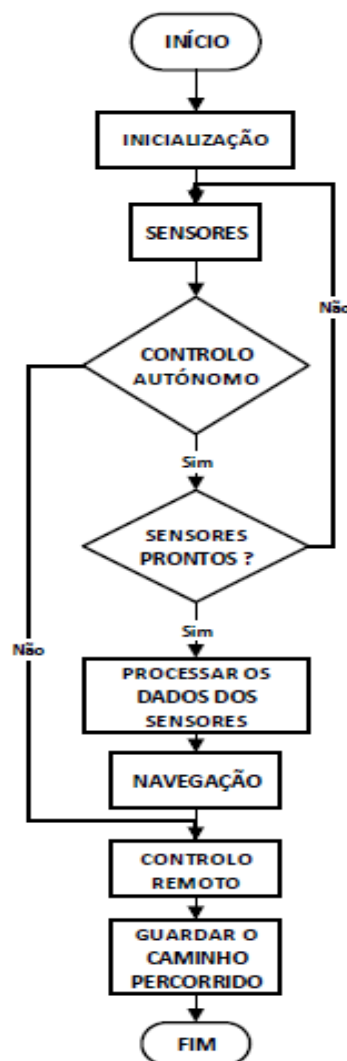


Fig. 41: telecomando para fazer o controlo da embarcação

3.2. Algoritmo usado no sistema de comando e controlo

Esta secção explicar o sistema de controlo apresentada a e posteriormente é dos blocos.

A estrutura de camadas, de onde navegador, que por composta por várias por determinar a camada controlo colocar os motores de direção desejada.



tem como objetivo funcionamento do automático. É arquitetura do sistema, especificado cada um baseia-se num princípio se destacam as camadas sua vez também camadas, responsável direção “ideal”, e a remoto, responsável por forma a alcançar a

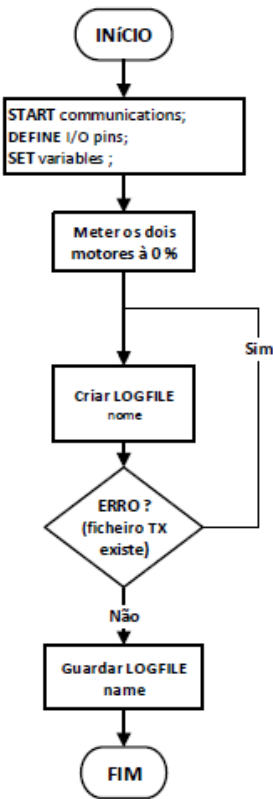
Fig. 42: Fluxograma geral do sistema de controlo autónomo

Neste fluxograma acima da fig.53 mostra-se os blocos base no qual o sistema se divide. Considere-se os seguintes blocos:

- Inicialização de navegação;

3.2.1. Inicialização do

O bloco da importante na fase permitir correr o resto tem que declarar todas bloco acima referido comunicações série



- do sistema;
- Processamento de dados

sistema inicialização tem um papel inicial do programa. Este bloco vai do programa, sendo que para tal as variáveis necessárias para cada no fluxograma geral, inicializar as dos diversos sensores e, não menos

importante, criar um ficheiro onde é guardado o trajeto que a embarcação faz num cartão SD. Além disso este bloco irá pôr os dois motores à 0 % (alavanca ao meio), senão depois quando for preciso pôr motores a trabalhar noutra regime não responderão.

Fig. 43: Fluxograma do bloco "inicializações"

O bloco " inicializações "foi dividido em rotinas, representando cada atividade uma rotina.

3.2.2. Processamento de dados de navegação

Depois de estabelecida a comunicação com os sensores, e garantir que já é possível avançar para a parte da exploração desses dados e consequente condução da embarcação, torna-se necessário realizar o processamento destes. Assim, este bloco tem como objetivo utilizar os dados dos sensores, como a posição GPS e a proa da embarcação, para fazer as respetivas comparações e enviar as ordens necessárias para os motores, tal como demonstrado nos seguintes fluxogramas.

É neste bloco que o programa com os dados das referências de posição, vai calcular a distância à posição desejada, e o seu azimuth. Consoante a distância à posição desejada este bloco vai atuar nos motores.

No bloco de baixo o algoritmo começa por calcular a distancia até ao próximo wp, de seguida compara esta distancia com um raio pré definido que determina a chegada ao Wp, se essa distancia for menor que o tal raio, que dizer que a embarcação encontra-se já no ponto pretendido, assim irá incrementar o contador dos Wp se houver mais WP e continua a missão, caso não houver mais WP, manda parar os motores. Este bloco é considerado o principal dos blocos da navegação.

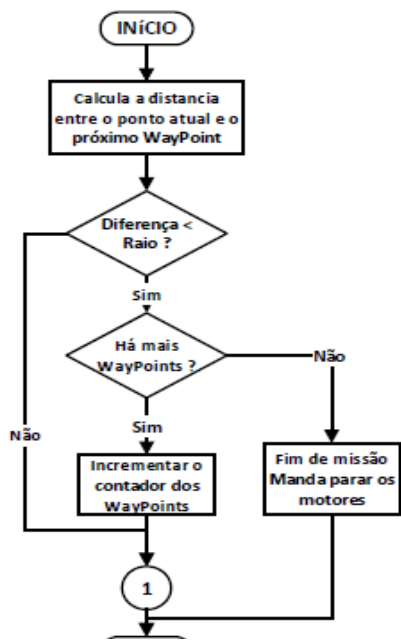
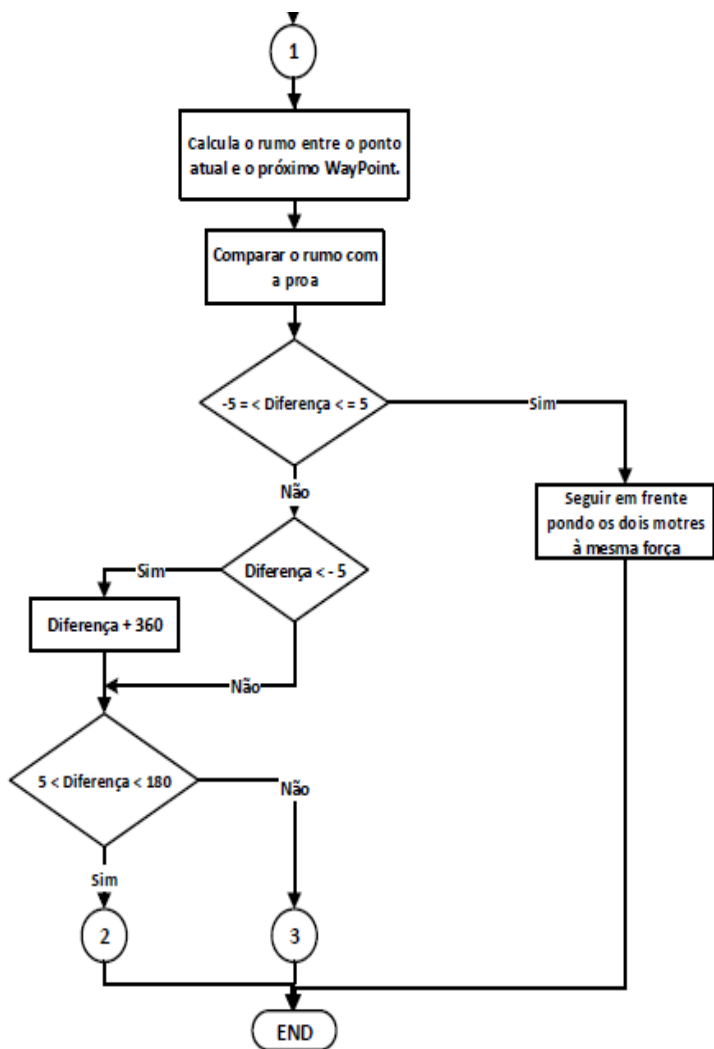


Fig. 44: Bloco de processamento de dados e navegação 1



termos a
queremos
seguinte
rumo até
chegar a

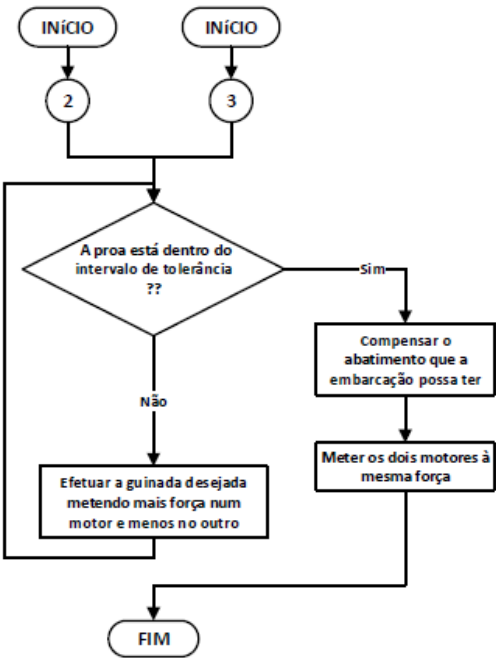
rumo, e
sempre
se for
tivermos
que

Depois de
distância até onde
ir, o objetivo do
bloco é calcular o
esse ponto, pois para
esse ponto é
necessário ter um
precisa de ser
corrigido e alterado
preciso também caso
condições adversas
implicam fazer isso.
Embarcação, se a

diferença estiver dentro do intervalo de tolerância manda seguir para frente, se não manda fazer as respectivas guinadas consoante a relação entre a proa e rumo.

Fig. 45: Bloco de processamento de dados e navegação 2

Apos saber preciso guinar, agora atuar nos motores fazer a respetiva travar o seguimento ter, o algoritmo do lado contrario segundos, depois do intervalo de



para que direção é o algoritmo vai mandar por forma diferencial para guinada. De seguida para que a embarcação possa manda também o motor atuar durante alguns compara se está dentro tolerância e manda seguir

para frente.

Fig. 46: Bloco de processamento de dados e navegação 3

De seguida será explicado a forma como foram feitos os diversos processos mencionados nos fluxogramas.

3.2.2.1. Calcular a distância entre pontos

Sabendo o *WayPoint* para onde a nossa embarcação tem que ir, e tendo as posições atuais fornecidas pelo GPS, agora necessitamos de saber qual é a distância entre estes dois pontos para conseguir parar ou continuar para o próximo *Waypoint* quando esta última se aproximar de zero ou de um raio de tolerância pré-definido.

Para calcular a distancia entre a posição atual dada pelo GPS (latitude e longitude) e o próximo *WAYPOINT*, podemos usar a fórmula de *haversine*¹⁴. A fórmula de *Haversine* é uma equação usada em navegação, fornecendo distâncias entre dois pontos (d) de uma esfera de raio R, a partir de suas latitudes (ϕ) e longitudes (λ). A equação é então dada por:

:

$$hav\sin\left(\frac{d}{R}\right) = hav\sin(\Delta\phi) + \cos\phi_1 \cos\phi_2 hav\sin(\Delta\lambda)$$

$\Delta\phi = \phi_1 - \phi_2$; a diferença entre as latitudes dos dois pontos

¹⁴ A fórmula de *Haversine* é uma equação usada em navegação, fornecendo distâncias entre dois pontos (d) de uma esfera de raio R, a partir de suas latitudes (ϕ) e longitudes (λ).

A equação é então dada por: $hav\sin\left(\frac{d}{R}\right) = hav\sin(\Delta\phi) + \cos\phi_1 \cos\phi_2 hav\sin(\Delta\lambda)$

$\Delta\lambda = \lambda_1 - \lambda_2$; a diferença entre as longitudes dos dois pontos

R ; Raio da esfera (globo terrestre)

Ao substituírmos *Haversine* (d/R), dado acima por h . Podemos obter d tanto pela simples aplicação da *Haversine* inversa, quanto pelo uso da função arcosseno (inverso do seno):

$$d = R. \text{havversin}^{-1}(h) = 2R. \arcsin(\sqrt{h})$$

Substituindo h :

$$d = R. \text{havversin}^{-1}(h) = 2R. \arcsin(\sqrt{\text{havversin}(\Delta\phi) + \cos \phi_1 \cos \phi_2 \text{havversin}(\Delta\lambda)})$$

Sendo que:

$$\text{havversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$$

Temos:

$$d = R. \text{havversin}^{-1}(h) = 2R. \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

3.2.2.2. Cálculo do rumo

Agora temos a distância entre a embarcação e o *WayPoint* pretendido, mas para conseguir fazer chegar a embarcação até esse *WayPoint*, esta última precisa de um rumo a seguir para chegar a esse *WayPoint* sem se perder. Este rumo é calculado através da equação referida em baixo.

$$\theta = \text{atan2}(\sin\Delta\lambda \cdot \cos\phi_2, \cos\phi_1 \cdot \sin\phi_2 - \sin\phi_1 \cdot \cos\phi_2 \cdot \cos\Delta\lambda)$$

Se pusermos:

$$y = \sin\Delta\lambda \cdot \cos\phi_2, \text{ e } x = \cos\phi_1 \cdot \sin\phi_2 - \sin\phi_1 \cdot \cos\phi_2 \cdot \cos\Delta\lambda$$

Temos:

$$\theta = 2\text{atan}\left(\frac{\sqrt{x^2 + y^2} - x}{y}\right)$$

θ ; rumo entre dois pontos geográficos

$\Delta\phi = \phi_1 - \phi_2$; a diferença entre as latitudes dos dois pontos

$\Delta\lambda = \lambda_1 - \lambda_2$; a diferença entre as longitudes dos dois pontos

É de salientar que o que foi descrito em cima só serve se a nossa embarcação não tiver em condições ambientais adversas (corrente e/ou vento) que irão implicar ao algoritmo

estabelecer um novo para compensar o abatimento provocado por estes últimos. Este novo rumo será necessário se o algoritmo detetar que a embarcação está sempre a fugir do rumo principal de um lado ou de outro, assim o algoritmo irá impor um novo rumo no lado inverso ao abatimento para que a nossa embarcação siga o rumo desejado teoricamente e evitando andar de forma harmónica com períodos pequenos e amplitudes grandes como mostram as figuras em baixo.

O algoritmo reconhece este novo rumo, depois de fazer um determinado número de guinadas e a embarcação continua a voltar para a mesma proa que o algoritmo tentou evitar. Assim a decisão será estabelecer um novo rumo que é igual ao antigo rumo (+ ou -) a diferença entre o rumo provocado pelas condições ambientais e o rumo desejado (θ) ver fig. Em baixo.

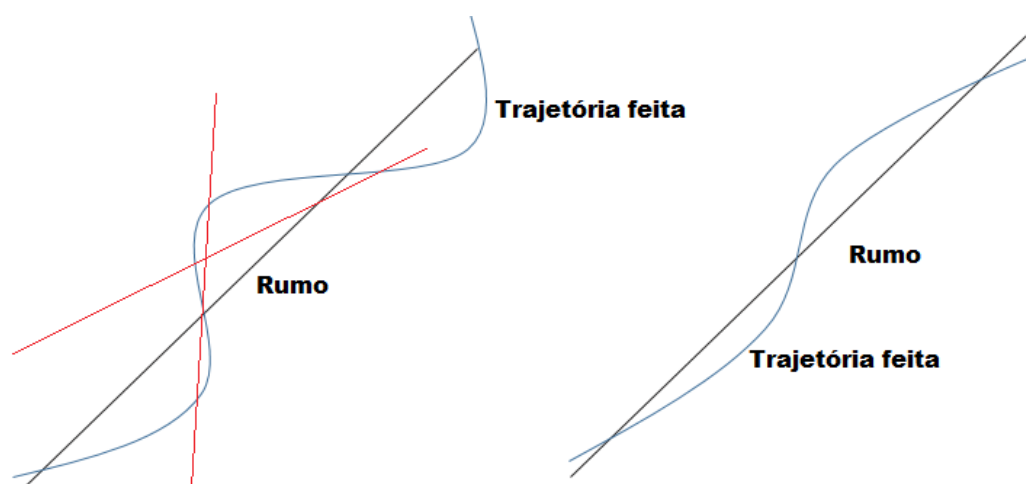


Fig. 47: Trajetórias a evitar

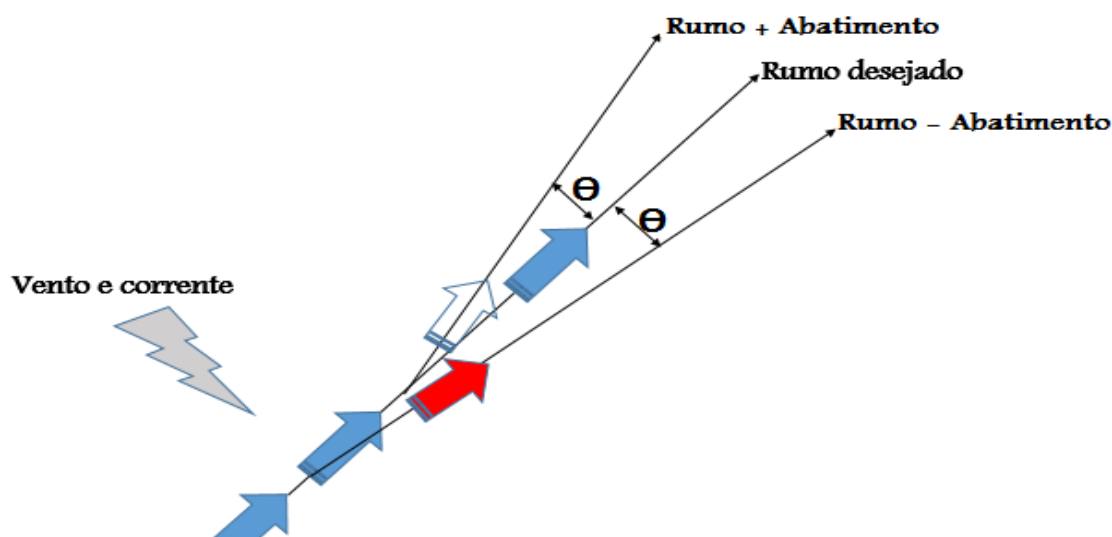


Fig. 48: como o algoritmo recalcula um novo rumo para a embarcação

3.2.2.3. Controlo de velocidade dos motores

Uma vez temos a proa fornecida pela bússola e o rumo que foi calculado, podemos seguir para o próximo passo que consiste em comparar a nossa proa com o rumo; para saber para que lado temos virar a embarcação. Não precisamos de virar 270 graus a bombordo se podemos virar só 90 graus a estibordo. A figura em baixo mostra todas as possibilidades resultantes da comparação e o lado da guinada correspondente.

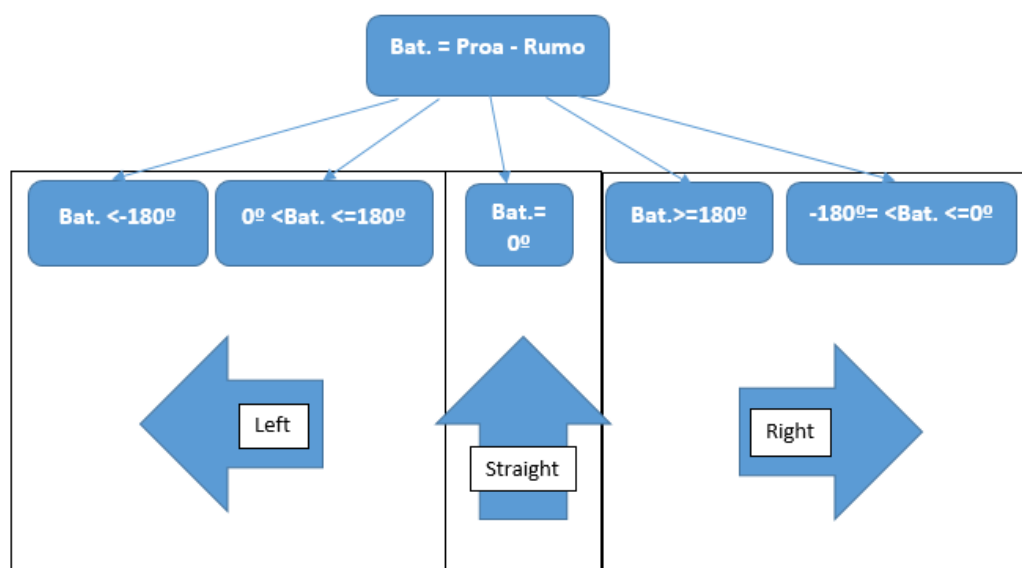


Fig. 49: Fluxograma da comparação

3.2.3. Algoritmo de navegação

Sabendo o bordo para onde temos de guinar, de seguida será estabelecido um Loop de guinada com um intervalo de tolerância de 5 graus para EB e para BB do rumo da embarcação, caso a nossa proa esteja dentro desse intervalo $[\text{rumo}-5, \text{rumo}+5]$, não vai ser preciso acionar os motores para guinar.

As guinadas serão feitas usando os dois motores, colocando um deles a funcionar com mais potência do que o outro, ou um a andar para vante e o outro para ré. Tudo isso será experimentado nos testes que se encontram descritos no capítulo seguinte, e os resultados de cada uma das possibilidades serão analisados consoante a sua eficiência, de modo a ser obtida a melhor guinada para a embarcação que será usada.

Também será preciso estabelecer um raio de distância dentro do qual se pode considerar a chegada ao waypoint feita e passar para o próximo waypoint ou parar a embarcação.

Os dados relacionados com a proa da embarcação e da distância até a chegada precisam de ser atualizados de forma constante, e em intervalos de tempo curtos, sobre tudo na altura em que algoritmo precisa de tomar decisões devido à deriva que possa prejudicar o desempenho do algoritmo e fazer com que a embarcação fuja dos limites sem o algoritmo reparar.

Ver ANEXO C; o algoritmo funcional.

4.1. Introdução

De forma a confirmar o desempenho do sistema foram efetuados vários testes. Numa primeira fase os testes foram realizados em ambiente seco, onde foram simuladas algumas condições que são possíveis de serem observadas quando a embarcação for para água, posteriormente as condições de teste foram sendo mais adversas em ambiente real ou testes da navegação da embarcação. O objetivo dos testes da primeira fase é para testar e afinar o algoritmo de comando e controlo antes de ir para a água, ou seja, deixar o algoritmo totalmente afinado sem nenhum problema e nenhuma anomalia que possa provocar um mau desempenho durante os testes na água, ou uma má interpretação dos resultados obtidos.

Excluem-se destes testes os realizados em laboratório para confirmar o correto funcionamento e interligação entre sistemas.

4.2. Simulação de condições de navegação

Neste subcapítulo vai-se falar sobre vários tópicos por forma a descrever de forma sucinta o cenário dos testes realizados, e efetuar uma boa análise dos resultados obtidos.

Condições e objetivos dos testes 1ª serie

Antes de iniciar os testes na água, onde as alterações e a monitorização em tempo real da embarcação são difíceis de realizar, optou-se por fazer ensaios em ambiente mais controlado. Deste modo foi feito um teste em ambiente seco sendo realizado na parada da escola naval. Não obstante a isso, tentou-se que as condições de teste fossem as mais aproximadas às de um teste no mar. Assim, o objetivo desta experimentação consiste em fazer uma primeira análise ao comportamento do sistema de controlo.

Cenário do teste

Depois de retirar os pontos GPS que correspondem aos WayPoints do percurso pretendido, e de introduzi-los no algoritmo, definindo a sua ordem, é possível visualizar o comportamento desse ultimo, consoante vários fatores dos quais: a posição do protótipo até ao WP, e a relação entre a proa e o rumo calculado.

Ao oscilar o protótipo de um lado para outro em relação ao rumo, conseguiu-se observar a reação dos motores. Também ao entrar dentro do raio da chegada ao WP foi possível ver a reação do algoritmo a incrementar o contador dos WP's e a calcular o novo rumo como demonstrado na análise dos resultados obtidos.

Esquemático utilizado

Material utilizado:

- *Arduíno*;
- Módulo GPS;
- Bússola magnética;
- Cartão SD com o *Shield* utilizado;
- Computador portátil;

Aparato experimental

Os vários sensores mencionados em cima foram conectados ao Arduíno com o propósito de fazer um protótipo prático e fácil de ser transportado e manuseado para simular as guinadas como se encontra demonstrado na Figura 50.

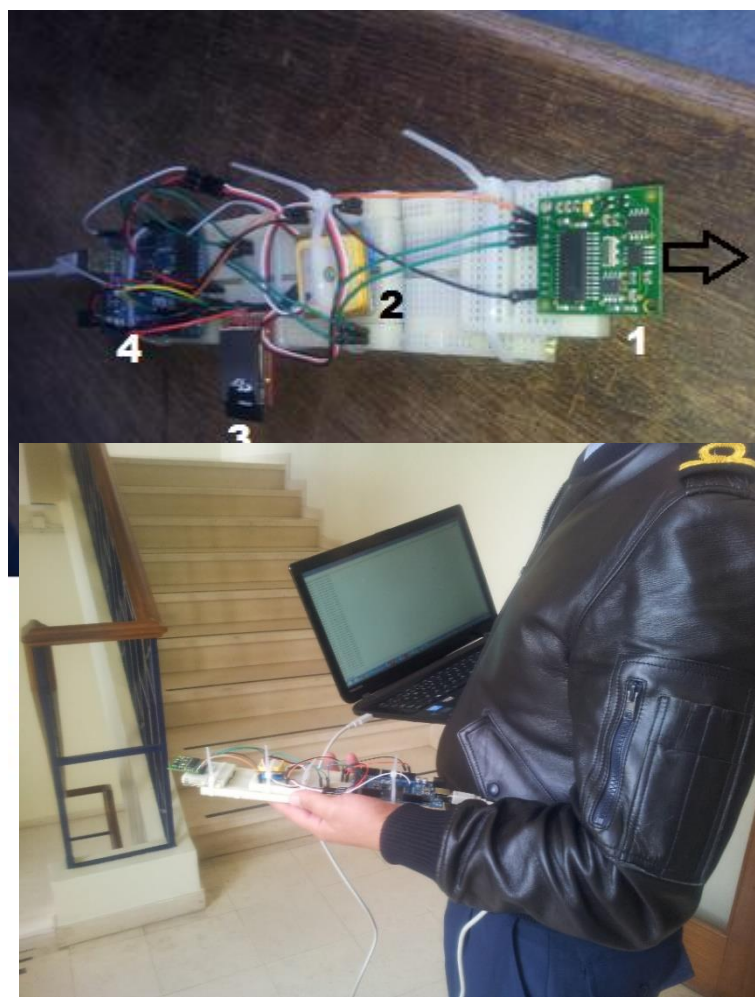


Fig. 51: Durante a realização do percurso

Descrição dos testes efetuados 1ª serie

Todo o sistema foi montado à semelhança do que aconteceria num teste real, ou seja, em ambiente aquático. Para tal foi efetuada a montagem de todo o *Hardware*, à exceção das placas de interface dos motores, sendo o seu funcionamento simulado e monitorizado no computador, através de um código elaborado no Arduino.

Por forma a simular as guinadas e as condições adversas, tais como o vento e a corrente que poderiam mover a embarcação, optou-se por realizar oscilações com a bússola nas direções pretendidas.

Relativamente ao percurso, foram definidos três WP's para serem percorridos como ilustrado na fig. 52; é de salientar que durante o teste o protótipo foi transportado à mão.



Fig. 52: WP's e o percurso

Condições e objetivos dos testes 2ª serie

O objetivo das experimentações da primeira serie consiste em fazer uma primeira análise ao comportamento e ao funcionamento do sistema de controlo sem ter em conta as condições ambientais adversas que possam impedir para que a nossa embarcação chegue ao local desejado, por isso, nos testes da 2ª serie pretendemos simular estas condições adversas ou alteração do rumo. Ou seja verificar se o nosso algoritmo realiza as correções necessárias como foi explicado na sub-subsecção (Cálculo do Rumo) do capítulo 3, e está demonstrado na fig48.

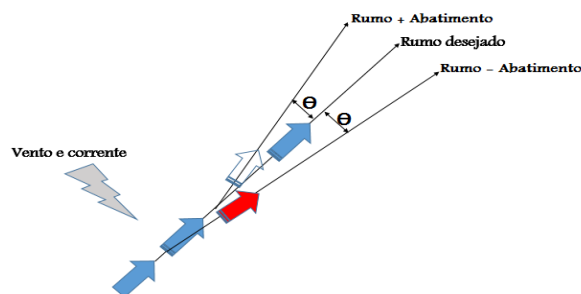


Fig. 53: como o algoritmo recalcula um novo rumo para a embarcação

Descrição dos testes efetuados 2ª serie

Este teste consiste como já foi referido anteriormente, a correção do rumo, ou seja, como demonstra a figura em baixo, a partida será o WP1, o objetivo é ir ao WP2, mas devido a condições adversas a embarcação fica sempre com abatimento na direção do WP3. O objetivo é ver o algoritmo a reagir e alterar o rumo como está explicado anteriormente.



Fig. 54: Trajetória do teste da 2ª serie

Análise dos resultados obtidos

A análise dos resultados vai ser feita separando os vários casos:

1º Caso:

Quando a diferença entre o rumo e a proa indica a necessidade de guinar para BB para entrar no angulo de tolerância e seguir em frente. Os motores atuam conforme pretendido metendo o motor de BB a 25% e o a de EB a 50%. Figura em baixo.

```
distancia atÃ© ao prÃ³ximo WP :
71.86
rumo entre o ponto atual e o prÃ³ximo WP
128.16
compass heading
137.00
motor 2 a 25% Forca a vante
motor 1 a 50% Forca a vante
Lat/Long(10^-5 deg): 38662445, -9147658 Fix age: 39469ms.
Lat/Long(float): 38.66244, -9.14766 Fix age: 39477ms.
Stats: characters: 3996 sentences: 1 failed checksum: 6
```

Fig. 55: Embarcação a guinar para BB

2º Caso:

Quando a diferença entre o rumo e a proa indica a necessidade de guinar para EB para entrar no angulo de tolerância e seguir em frente. Os motores atuam conforme pretendido metendo o motor de BB a 50% e o a de EB a 25%. Figura

```
rumo entre o ponto atual e o prÃ³ximo WP
130.52
compass heading
89.00
motor 2 50% Forca a vante
motor 1 25% Forca a vante
Lat/Long(10^-5 deg): 38662512, -9147700 Fix age: 10836ms.
Lat/Long(float): 38.66251, -9.14770 Fix age: 10843ms.
Stats: characters: 31470 sentences: 43 failed checksum: 83
distancia atÃ© ao prÃ³ximo WP :
79.51
```

Fig. 56: Embarcação a guinar para EB

3º Caso:


```
motor 1 50% Forca a vante
motor 2 50% Forca a vante
motor 1 50% Forca a vante
motor 2 50% Forca a vante
Lat/Long(10^-5 deg): 38662538, -9147655 Fix age: 4ms.
Lat/Long(float): 38.66254, -9.14765 Fix age: 6ms.
Stats: characters: 361 sentences: 1 failed checksum: 3
distancia at   ao pr  ximo WP :
1.37
Motor 1 Parado
Motor 2 Parado
mission done
```

Fig. 59: Algoritmo a mandar parar os motores

O processo do *refresh* do algoritmo   feito a cada 5 segundos de maneira a conseguir, de uma forma precisa, estar sempre atualizado em rela  o ao que acontece durante a navega  o.

Conclus  es e considera  es

Salienta-se que foram efetuadas v rias tentativas, at  se conseguir efetuar o percurso com sucesso. Este problema deveu-se   exist ncia de erros no algoritmo de navega  o e na liga  o dos sensores, s  detetados em testes posteriores.

O algoritmo inicial sofreu v rias altera  es conforme as tentativas realizadas, e as exig ncias pretendidas.

O objetivo principal foi cumprido assim possibilitando a passagem para os testes em ambiente real, n o esquecendo a necessidade de cobrir o prot tipo para n o ser tocado pela  gua, este ultimo que vai ser refor ado com as placas de interface e as respetivas baterias. Tamb m ter muito cuidado com o problema das falhas de comunica  es que poder  dificultar o desempenho do algoritmo.

4.3. Testes de navega  o da embarca  o

Depois de ter o algoritmo de controlo afinado ao m ximo que os testes simulados permitiram, o objetivo dos testes na  gua   realizar realmente a trajet ria pretendida com as condi  es adversas presentes, e com os motores a fazerem movimentar a embarca  o de forma diferencial.

Preparações para os testes

Para ter a embarcação pronta era preciso fazer uma serie de preparações, algumas delas encontram-se demonstradas nas seguintes figuras:



Fig. 60: Embarcação no seu estado inicial



Fig. 61: Embarcação já com motores montados

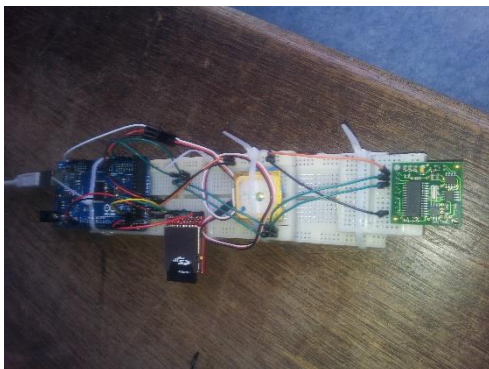


Fig. 62: Protótipo de controlo Arduino + Sensores



Fig. 63: Posição do protótipo na embarcação e na caixa com o resto dos equipamentos



Fig. 64: Alimentação do Arduino



Fig. 65: Alimentação dos motores

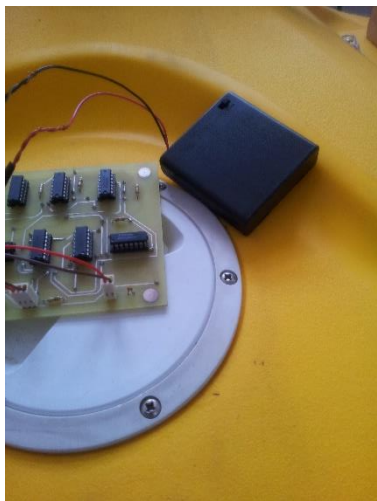


Fig. 66: Alimentação das placas de interface

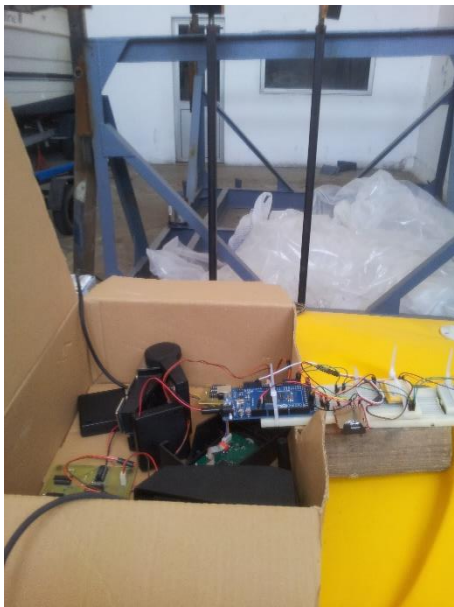


Fig. 67: A caixa onde se interligam todos os equipamentos

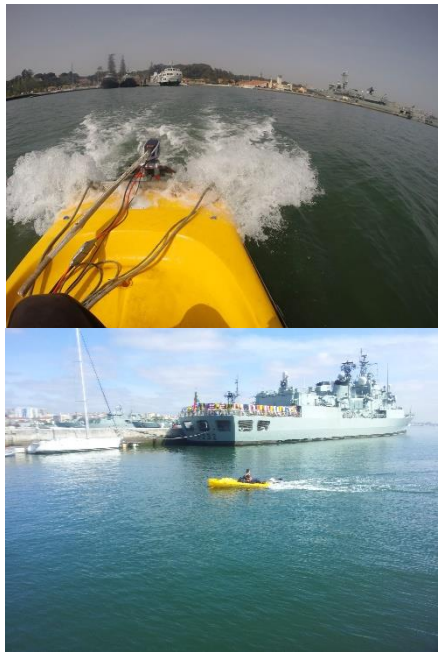


Fig. 68: Embarcação na água

De acordo com o que está descrito em cima nas fotos referidas á preparação da embarcação para realizar os testes de navegação podemos dizer o seguinte:

- Foi usado um conjunto de 4 pilhas de 1.5 Volts para alimentar tanto o Arduíno como as placas de interface.
- Foram usadas duas baterias de 12 volts em serie para alimentar os motores.
- O protótipo que contém o Arduíno e os sensores foi colocado numa posição avante da embarcação para conseguir estar longe de todos os impedimentos encontrados na embarcação.
- O material eletrónico foi colocado numa caixa e foram feitos dentro dessa caixa todas as ligação entre o Arduíno e as placas de interface, e também entre as placas de interface e os microcontroladores dos motores. Essa caixa foi coberta com plástico para isolá-la da água.

Condições e objetivos dos testes

Cenário do teste

O cenário destes testes consiste em realizar uma trajetória na água, esta trajetória composta por quatro pontos. O objetivo é fazer a tal trajetória passando por esses pontos e realizar trajetória de forma ideal, ou seja, uma trajetória harmónica com amplitude mais baixa possível.

Depois de retirar os pontos GPS que correspondem aos WayPoints do percurso pretendido, e de introduzi-los no algoritmo, definindo a sua ordem, a seguir, é colocar a embarcação no WP número 1, e deixar a embarcação continuar o resto da trajetória de forma autónoma. A figura em baixo mostra a trajetória escolhida para fazer este teste. Foi escolhida devido ao fato de estar numa zona bem escondida das condições ambientais adversas.

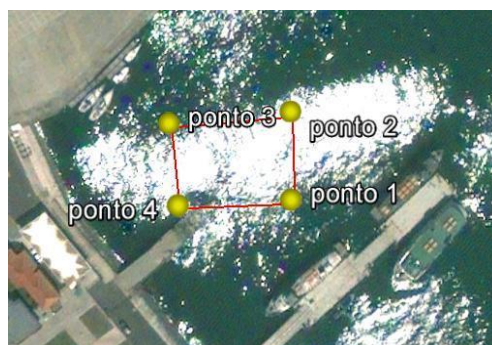


Fig. 69: Trajetória escolhida para fazer os testes

Análise dos resultados obtidos

Devido a vários problemas encontrados durante a realização dos testes, falhas e maus contactos. Os resultados obtidos não foram os que estivemos a espera. Fazendo uma trajetória completamente aleatória em vez de seguir os quatro wps.

Conclusões e considerações

Fica para fazer

5.1.Introdução

Neste capítulo pretende-se fazer um resumo do trabalho desenvolvido ao longo do período que foi designado para o desenvolvimento desta tese de mestrado. Neste capítulo será feita uma revisão dos objetivos traçados no início deste trabalho, dos alcançados e dos que ficaram por alcançar. Pretende-se também tirar conclusões detalhadas acerca do projeto e deixar algumas sugestões para o futuro com o intuito de desenvolver e melhorar este projeto de comando e controlo de motores elétricos.

3.2.Revisão dos objetivos traçados

Inicialmente foram propostos os seguintes objetivos:

- Fazer uma seleção e efetuar uma pesquisa sobre os estudos realizados no âmbito dos sistemas de comando e controlo diferencial usando motores elétricos, e dos equipamentos necessários criar esses sistemas de comando e controlo, em modo autónomo, e em modo remoto.
- Fazer a placa de interface entre os motores elétricos e o *Arduíno*.
- Controlar as velocidades dos motores com o *Arduíno*.
- Programar um sistema de comando e controlo diferencial que funcione em dois modos de funcionamento (remoto, autónomo).
- Efetuar testes simulados para fazer uma primeira análise ao comportamento do algoritmo e do sistema.
- Fazer testes de navegação da embarcação (em ambiente real).

O primeiro objetivo faz referência a todo o estudo e soluções relativas ao sistema que realizado. O Capítulo 2, Estado da Arte e conceitos gerais do sistema, é onde se insere esse objetivo. Neste Capítulo é feito um estudo sobre os diversos sistemas de comando e controlo

autónomo, sobre algoritmos de controlo e relativamente ao *Hardware* que foi utilizado. Foi feito uma síntese dos estudos relacionados com o tema do projeto, essa síntese foi explorado no final deste mesmo capítulo para definir os requisitos funcionais do sistema desenvolvido (requisitos de software e requisitos de hardware).

Em relação ao segundo, terceiro e quarto objetivo, estão descritos no Capítulo 3, Arquitetura do Sistema e comando e controlo. Este capítulo é dividido em três partes, Interface, Hardware e Software. É no subcapítulo do Software que é descrita a programação do sistema de comando e controlo com ambos os modos de funcionamento. No subcapítulo Hardware é onde estão descritos todos os sensores utilizados e a forma como funcionam e interagem com o Arduíno, e também, todas as ligações físicas efetuadas. No Interface está apresentado como se consegue controlar as velocidades dos motores elétricos, e a forma como são enviadas e armazenadas as palavras relativas às velocidades, do Arduíno para a placa de interface onde posteriormente o PIC dos motores irá buscar.

Os últimos objetivos, e não menos importantes, estão relacionado com os testes ao projeto. Está presente no Capítulo 4, Testes e resultados, Simulados e em ambiente real, não só os testes simulados que foram utilizados para testar o algoritmo desenvolvido, como também testes realizados na água. Os testes simulados foram feitos para testar e afinar o algoritmo de comando e controlo antes de ir para a água.

O protótipo utilizado nestes testes em campo é leve, pequeno, e fácil de transportar. Utiliza também dois motores elétricos, tal como idealizado inicialmente. No entanto, tem algumas desvantagens pelo facto de ser frágil, e providenciar muitos maus contatos e falha de contatos durante os testes que dificultou de forma considerável o bem cumprimento dos mesmos. Também a embarcação utilizada para fazer os testes na água não foi a mais adequada ao protótipo desenvolvido e para os motores usados, pois a embarcação não aguenta com o peso dos dois motores ficando com a parte de ré dela muito submersa na água

dificultando a movimentação da embarcação e fez com que um dos motores ficasse estragado devido a entrada da água a parte da eletrónica dele, fazendo com o trabalho parasse durante muito tempo até que foi recuperado no departamento de armas e eletrónica da escola naval.

3.3. Desenvolvimentos futuros

Com o desenvolvimento do projeto muitos requisitos se tornam mais evidentes. Segue abaixo mencionado os desenvolvimentos futuros que o autor acha mais relevante.

- Resolver o problema de maus contatos que fiz com o trabalho parasse muitas vezes.
- Desenvolver o *software* e hardware do protótipo desenvolvido por forma a acrescentar os seguintes modos de funcionamento que são também imprescindíveis para o bom cumprimento das missões pré-definidas no projeto Icárus e busca e salvamento no mar:
- **Modo Externo** que consiste numa operação autónoma com uma missão remotamente programada em tempo real.
- **Modo de Âncora** que permite manter uma estação em execução, em que a embarcação mantém a sua posição compensando as flutuações causadas pelos ventos, correntes de ar ou outras influências.
- **Modo de Controlo Remoto** em que a embarcação pode ser remotamente controlada pelo utilizador. Este modo foi referido e estudada neste projeto, mas devido á alguns problemas encontrados com o telecomando remoto, não foi realizado à 100%. É preciso que se adquira um telecomando novo, fazer testes com os motores para afinar bem a parte o algoritmo responsável por esta parte e integra-lo de forma eficiente em todo o *software*, que irá implicar mudanças no algoritmo atual feito neste projeto.
- **Modo Negligente** onde a atuação da embarcação é cortada, fazendo com que flutue de acordo com as perturbações externas (ventos e correntes).
- Adquirir uma embarcação mais adequada, com uma estrutura e arquitetura bem estudadas consoante todos os requisitos necessários tendo em conta o peso dos motores, do protótipo e o peso da balsa salva vidas que será lançado quando a embarcação chega ao sítio do desastre. Ambas a arquitetura e a estrutura desta ultima

foram estudadas na tese do Aspirante EN-MEC Brahim Younes. A embarcação utilizada atualmente é uma embarcação aleatória que proporcionou imensas dificuldades para realizar os testes de navegação, sendo ela, com os motores a funcionarem em regimes médio até altos, a parte de ré dela fica completamente imersa dentro da água. Que provocou por sua vez danos nos motores e no protótipo utilizado.



Fig. 70: Exemplo de embarcações usadas no mesmo âmbito de trabalho (Projeto Icarus)

- Adquirir material e ferramentas mais adequadas para realizar o projeto com maior taxa de sucesso e com criatividade mais elevada.

Referências Bibliográficas

- [1] K. Ogata, Engenharia de Controlo Moderno, 5ª Edição ed., Pearson Education Inc., 2003.
- [2] F. Pinto, Sistemas de Automação Controlo, CETEC, Ed., 2005.
- [3] L. Baptista, Controlo de Sistemas, Escola Superior Náutica Infante D. Henrique.
- [4] L. Baptista, Instrumentação e Controlo, Lisbon: Escola SuperioeNáutica Infante D.Henrique, 2012.
- [5] C. Johnson, Controlo de processos – Tecnologia da Instrumentação, Fundação Calouste Gulbenkian, 1990.
- [6] L. Sghieri e A. Nishinari, Controlo Automático de Processos Industriais-Instrumentação, Edgard Blucher, Ida, 1973.
- [7] M. Babb, "Pneumatic Instruments Gave Birth to Automatic Control," *Control Engineering*, Outubro 1990.
- [8] "How Differential Gear works (BEST Tutorial)," [Online]. Available: <https://www.youtube.com/watch?v=K4JhruinbWchttp>.
- [9] D. king-Hele, "Erasmus Darwin's Improved Design for Steering Carriages," *The Royal Society*, April 2008.
- [10] L. Conrad, "Virtual Skyes - NASA: <http://virtualskies.arc.nasa.gov/index.html>," Abril 2010. [Online]. [Acedido em May 2015].
- [11] J. P. Snyder, em *A Working Manual. U.S. Geological Survey Professional*, Washinton, DC:, USA, 1987, pp. Paper 1395 (pp. 90-91)..
- [12] W. G. S. K. M. & K. H. Gellert, Spherical Trigonometry. In *VNR Concise Encyclopedia of Mathematics*, 2nd ed (pp. 261-282) ed., New York: NY: Van Nostrand Reinhold., 1989.
- [13] R. W. Sinnott, Virtues of the Haversine. *Sky and Telescope* 68, p. 159., 1984.
- [14] "Calculate distance, bearing and more between Latitude/Longitude points.," [Online]. Available: <http://www.movable-type.co.uk/>. [Acedido em 26 06 2015].
- [15] "Development of an unmanned capsule for large-scale maritime search and rescue," FEUP, Porto, 2013.
- [16] Y. E. Zhao e J. Zhang, "Modeling and simulation of electronic differential system for an electric vehicle with two-motor-wheel drive," *IEEE*, Xian, China, 2009.
- [17] M. Rengarajan e G. Anitha, "Algorithm development and testing of low cost waypoint navigation system," Anna University, Chennai India, 2013.
- [18] M. Shashidhar, "Development of low cost controller and navigation system for unmanned ground vehicle, Dissertação de mestrado em engenharia elétrica (Master of Science in electrical engineering),," West Virginia University, EUA, 2008.
- [19] "Movable Type," [Online]. Available: <http://www.movable-type.co.uk>. [Acedido em 26 06 2015].

Este anexo consiste no relatório do trabalho que foi feito no âmbito do estágio de dois alunos da universidade nova de Lisboa, Faculdade de Ciências e Tecnologia. Neste estágio, aproveitou-se o trabalho desenvolvido na tese do 2TEN EN-AEL Jorge Jesus para fazer uma placa de interface entre o motor (PIC) e o *Arduíno*, também, foi descoberta a reta de velocidade dos motores que servia para conseguir mandar as palavras em binário referentes a velocidade.

“Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer.” In www.arduino.cc

Qualquer sistema de vela autónoma necessita de um ou mais processadores para as tarefas de comando e controlo. De entre os muitos sistemas disponíveis no mercado.

Nesta secção é descrito um pouco do conceito do Arduino em termos de Hardware.

Introdução

O projeto Arduino começou em Ivrea, Itália em 2005 e que tinha como objetivo a criação de uma placa de desenvolvimento mais acessível, para ser usada em projetos de escolas. Os seus fundadores são Massimo Banzi e David Cuartielles, que deram o nome de um bar ao projeto, que quer dizer um “grande amigo” em italiano.

O projeto Arduino ganhou uma menção honrosa na categoria Comunidades Digitais em 2006, pela Prix Ars Electrónica.

O Arduino é muitas vezes designado como uma placa de desenvolvimento de Hardware ou de Software, mas é muito mais que isso porque, além de ser um grande sucesso é também apoiado por uma grande comunidade de utilizadores em todo o Mundo que se ajudam mutuamente num fórum específico (<http://arduino.cc/forum/>).

As razões do seu sucesso são:

- Baixo custo;
- Simplicidade de utilização;
- Poder ser utilizado em vários ambientes (Windows, Macintosh e Linux).
- O Arduino é uma ferramenta que permite fazer tudo o que a imaginação permita, uma vez que, se podem fazer upgrades à placa base através de shields (Arduino, 2005).

Hardware

Nesta subsecção vou comparar os dois modelos, em termos de *hardware*, disponíveis na Escola Naval o *Arduino Duemilanove* e o *Arduino Mega 2560*.

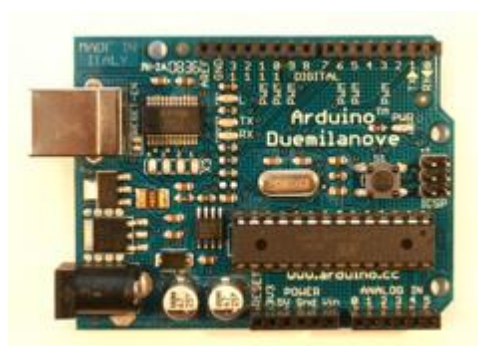


Fig. 71: Arduino Duemilanove

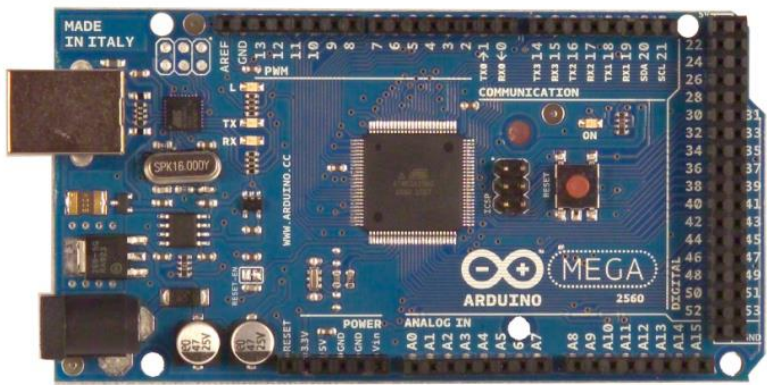


Fig. 72: Arduino Mega 2560

Microprocessador vs Microcontrolador

As diferenças entre um microprocessador e microcontrolador são no aspeto da funcionalidade, pois para um microprocessador ser usado é preciso usar outros componentes, tais como memória e componentes para receber e enviar dados ou seja o microprocessador é o componente essencial. Enquanto que o microcontrolador é projetado de modo a ter todos os componentes necessários ao seu funcionamento já estejam integrados nele.

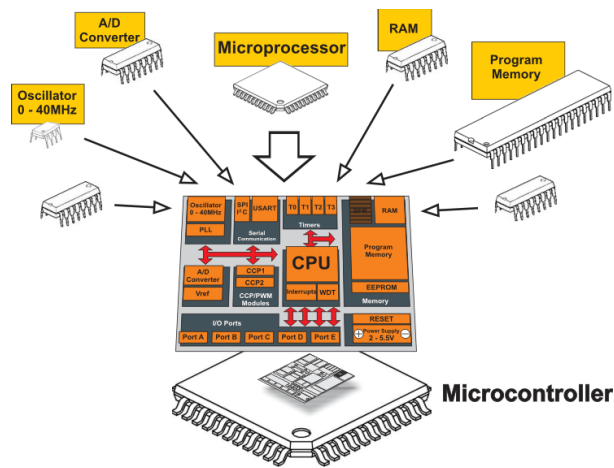


Fig. 73: Microcontrolador vs Microprocessador

Em suma, um microcontrolador é um microprocessador já integrado com dispositivos de memória, unidades de I/O e outros blocos de funções especiais (Figura 64).

O *Arduíno Duemilanove* tem como microcontrolador o ATmega168 enquanto o *Arduíno Mega* usa o ATmega2560.

Memória do Arduíno

Uma das principais diferenças entre os dois modelos apresentados é em termos de memória do microcontrolador utilizados. Sendo que os microcontroladores usados usam três tipos de memória:

Tabela 2: Tipo de Memória e utilização dada

FLASH	Onde o <i>bootloader</i> e o nosso programa é armazenado
SDRAM	Onde são guardadas e modificadas as variáveis ao longo da execução do programa
EEPROM	Pode ser usado para guardar informação (não volátil)

As Quantidades de memória disponíveis pelos modelos são:

Tabela 3: Quantidade de memória nos microcontroladores ATmega168 e Atmega2560

	ATmega168	ATmega2560
Flash	16KB	256KB
SRAM	1KB	8KB
EEPROM	512Bytes	4KB

Pela análise da tabela 2 podemos ver que o ATmega2560 é muito superior em termos de capacidade.

Pinout Arduíno

A outra grande diferença entre os dois modelos é em termos de pinout disponível. Existindo sete tipos de pinout disponíveis:

1)- Pinos de I/O digitais sendo que alguns tem a capacidade de criar sinais PWM. A capacidade de utilizar PWM, é muito importante pois permite a partir de um sinal que só pode assumir o estado lógico 0 (0V) ou 1 (5V) obter um sinal analógico.

2)-Pinos analógicos que para conseguirmos perceber os valores que obtemos temos que analisar a conversão A/D feita. O microcontrolador utilizado possui um conversor analógico de 10 bits.

3) Pinos de Ground

4) Pinos de 5 V

5) Pinos de 3.3 V

6) Pino de Analog Reference este pino permite mudar a referência analógica do standard 5V para o valor de entrada. Ficando todas as entradas analógicas com a referência introduzida.(Arduíno, 2005)

7) Pinos de reset que servem para fazer reset ao Arduíno.(Arduíno, 2005)

8) Pino de Vin pode servir para alimentar o Arduíno na gama de valores especificada ou, caso se alimente o Arduíno recorrendo a entrada USB ou a entrada tipo Jack, sendo que a alimentação do Jack recomendada e entre os 7 e os 12V, tem-se disponível no pino a tensão utilizada para alimentar o Arduíno (Arduíno, 2005).

Em termos de *pinout* dos modelos do *Arduíno* em estudo temos a seguinte tabela:

Tabela 4: PinOut dos Modelos Duemilinove e Mega2560

	Duemilinove	Mega2560
I\O digitais	14 (6 com PWM)	54 (14 com PMW)
Analógicos	6	16
Ground	3	5
5V	1	3
3.3V	1	1
Analog Reference	1	1
Reset	1	1

